LEVEL

12

ADA064613

KALMAN FILTERING AND SMOOTHING IN FOTONAP

For Orbit Determination Using GPS Measurements

September 1978
FINAL REPORT
ETL-0161

OLD DOMINION SYSTEMS, INC.
Gaithersburg, Maryland

79 02 12 027

ETL-0161

KALMAN FILTERING AND SMOOTHING IN FOTONAP

For Orbit Determination Using GPS Measurements

Georg E. Morduch
David A. Bergeron
Old Dominion Systems, Inc.
4 Professional Drive, Suite 119
Gaithersburg, Maryland 20760

September 1978

Final Report

Prepared for

U. S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 ETL-0161	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9 Final rept.
4. TITLE (and Subtitle) 6 KALMAN FILTERING AND SMOOTHING IN FOTONAP For Orbit Determination Using GPS Measurements.		5. TYPE OF REPORT & PERIOD COVERED Contract Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Georg E. Morduch, David A. Bergeron		8. CONTRACT OR GRANT NUMBER(s) 15 DAAK70-77-C-0254 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Old Dominion Systems, Inc. 4 Professional Drive, Suite 119 Gaithersburg, Maryland 20760		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060		12. REPORT DATE 11 September 1978
		13. NUMBER OF PAGES 129
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

orbit determination, Kalman filter, atmospheric refraction, atmospheric drag, Doppler tracking

390 661 fill

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Fotonap program has been modified to incorporate ~~(i)~~ A Kalman filter and a fixed lag smoother, ~~(ii)~~ the capability to handle GPS measurements through the filter/smoother, ~~(iii)~~ a Lockheed-Jacchia dynamic atmospheric model, ~~(iv)~~ a discretely changing atmospheric drag coefficient (drag segmentation), and ~~(v)~~ the capability to accept as input to the regular Fotonap the output from the filter/smoother including the full covariance matrix. The derivation of all the required equations is given,

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) in this report.

Updated versions of Fotonap exist for both CDC 6400 and Univac 1108 computers.

Preface. Authorization for implementing the changes to the Univac 1108 and CDC 6400 versions of Fotonap was given by the United States Army Engineer Topographic Laboratories, Fort Belvoir, Virginia under contract DAAK 70-77-C-0254. Mr. A. T. Blackburn served as Project Engineer for the government.

TABLE OF CONTENTS

TABLE OF CONTENTS CONTINUED

TABLE OF CONTENTS CONTINUED

1. Introduction

This report describes the mathematical analysis on which the program modifications to Fotonap are based. Also included in the report is some program documentation and a description of the final test runs used in the program check out. The program modifications may be subdivided into 7 separate groups, 6 of which are mathematical in nature and therefore described here. To deal with the added capabilities, the set of control cards used to run Fotonap has been augmented. A new Fotonap user's guide has been issued as a separate document. Program changes based on the analysis given in this report have been implemented and checked out on both the CDC 6400 and the Univac 1108 versions of Fotonap.

The most fundamental change to Fotonap is the inclusion of a Kalman filter and a fixed lag smoother. The smoother formulation is considerably more complicated than that of the filter. This applies both to the mathematical analysis and to the program. The core storage requirements are also much higher. The analysis is based on Gelb (1977), though checking back to the original source (Meditch, 1969) one of the required equations was found to be in error. The formulation implemented in Fotonap, however, differs slightly from that given by Meditch (1969). The Fotonap formulation, though mathematically equivalent, requires slightly less core storage. The filter and smoother analysis is given in Section 2. The description of the satellite drag coefficient and the GPS user clock as random variables is based on some equations given by Dr. Ballew (1977) of DMA Aerospace Center, St. Louis. This is given in Sections 3 and 4.

Formulae for calculating the state transition matrices (required by the filter/smoother) from those computed by the regular Fotonap integrator are given in Section 5.

A brief description of the Global Positioning System (GPS) is given in Section 6. A derivation of all the equations required for handling GPS measurements is also contained within this section. A considerable amount of effort was spent on developing an efficient scheme for selecting GPS satellites when running the program in a simulation mode. It had originally been thought possible to use the Morduch (1976) method, but that was developed for somewhat different requirements. The rather lengthy analysis is given in Appendix E.

Section 7 gives the mathematical analysis for drag segmentation, which is a scheme whereby the satellite drag coefficient changes discretely at fixed (by the program user) intervals of time. A novel (optional) feature of the implemented scheme is that the coefficients may be mutually constrained, the strength of the constraints being determined by the program user. (Very strong constraints effectively eliminate the distinction between the segments. This feature was utilized in the program check-out.)

A new atmospheric model, the Lockheed-Jacchia Atmosphere has been added to Fotonap. The program is a modification of one supplied by Mr. George Stentz (1978) of DMA Aerospace Center. A description of the formulae used is given in Section 8. Section 9 of the report gives a general description of all program changes. The changes to the Fotonap user's guide are indicated in Section 10. Section 11, the last section of the main text, describes some of the test runs made in checking out the new version of Fotonap.

A set of 7 appendices are also included in the report. The first 5 (A through E) give detailed derivations of some of the formulae used in the main text. Appendices F and G updates some of the previous program documentation (Hartwell, 1975) relating to Fotonap.

Fotonap is sometimes spelled Photonap in this report. It is the same program.

2. Mathematical Analysis For Fixed Lag Smoother

2.1 Definition of Terms

E Operator denoting 'Expected value of'

A^{-1} Inverse of matrix A

A^T Transpose of matrix A

$A^{-T} = (A^{-1})^T = (A^T)^{-1}$

Cov y Covariance of y

$\text{Cov } y = E(y-Ey)(y-Ey)^T$

$x(k,n)$ Estimate of parameter vector at time-point k after processing measurements at time-points 1 through n.

$x1(k) = x(k,k)$

$x2(k) = x(k+1,k)$

$xS(k) = x(1,k) \quad \text{if } k \leq L$

$\quad\quad = x(k-L,k) \quad \text{if } k > L$

L Lag constant

$b(k,n)$ Estimate of parameter vector at time-point k after processing measurements at time-points n through N and $n \geq k$

$xT(k)$ True parameter vector at time-point k

$\tilde{x}(k,n) = x(k,n)-xT(k)$

$\tilde{b}(k,n) = b(k,n)-xT(k)$

$P(k,n)$ Estimated covariance of $\tilde{x}(k,n)$

$P1(k) = P(k,k)$

$P2(k) = P(k+1,k)$

$PS(k) = P(1,k) \quad \text{if } k \leq L$

$\quad\quad = P(k-L,k) \quad \text{if } k > L$

$B(k,n)$ Estimated covariance of $\tilde{b}(k,n)$

$M(j,k)$ State transition matrix relating the parameter vectors (system states) at time-points j and k.

$M(k,j) = M(j,k)^{-1}$

$W(k)$ State noise. See equations (2.4.1) and (2.4.2)

$Q(k) = E\ W(k)W(k)^T$

$A(k) = P(k,k)M(k+1,k)^T P(k+1,k)^{-1}$

$C(j,k) = A(j)A(j+1)....A(k)$ [Defined only for $k \geq j$]

$C1(k) = C(h,k)$, where $h \equiv 1 \pmod L$ and $k-L<h \leq k$

$C2(k) = C(k+1-L,k)$ [Defined only for $k > L$]

$dP(k)$ Change in covariance of parameter estimate at time-point k [See Equations (2.2.6) and (2.2.7)]

$dx(k)$ Change in parameter estimate at time-point k [See equation (2.2.13) and 2.2.14)]

$\Delta P(k) = P(k-L,k) - P(k-L,k-L)$

$= PS(k) - P1(k-L)$

$\Delta x(k) = x(k-L,k) - x(k-L,k-L)$

$= xS(k) - x1(k-L)$

$G(k)$ Kalman gain matrix [See equation (2.2.8)]

$H(k)$ Matrix of partial derivatives of measurements with respect to parameters

$Z(k)$ Vector of measurements

r Measurement noise

$R = Err^T$

$F(k) = P(k,N) - P(k,N+1)$
$f(k) = x(k,N+1) - x(k,n)$

2.2 Filter and Smoother Equations

The following formulae will be derived.

$$P(k+1,k) = M(k+1,k)P(k,k)M(k+1,k)^T+Q(k), \qquad (2.2.1)$$

$$A(k) = P(k,k)M(k+1,k)^T P(k+1,k)^{-1} \qquad (2.2.2)$$

$$C(j,k) = C(k,k-1)A(k) \qquad (2.2.3)$$

$$C(j,k-1) = A(j-1)^{-1}C(j-1,k-1) \qquad (2.2.4)$$

$$C(j,j-1) = I, \quad \text{the identity matrix} \qquad (2.2.5)$$

$$P(k+1,k+1) = P(k+1,k) - dP(k+1) \qquad (2.2.6)$$

$$dP(k+1) = G(k+1)H(k+1)P(k+1,k) \qquad (2.2.7)$$

$$\begin{aligned} G(k+1) = {} & P(k+1,k)H(k+1)^T[R(k+1) + \\ & + H(k+1)P(k+1,k)H(k+1)^T]^{-1} \end{aligned} \qquad (2.2.8)$$

$$P(l,k+1) = P(l,k) - C(l,k)dP(k+1)C(l,k)^T \qquad (2.2.9)$$

$$\begin{aligned} P(k+1-L,k+1) = {} & P(k+1-L,k-L) + \\ & + A(k-L)^{-1}\Delta P(k)A(k-L)^{-T} \\ & - C(k+1-L,k)dP(k+1)C(k+1-L,k)^T \end{aligned} \qquad (2.2.10)$$

$$\Delta P(k) = P(k-L,k) - P(k-L,k-L) \qquad (2.2.11)$$

$$x(k+1,k) = M(k+1,k)x(k,k) \tag{2.2.12}$$

$$x(k+1,k+1) = x(k+1,k) + dx(k+1) \tag{2.2.13}$$

$$dx(k+1) = G(k+1)[Z(k+1)-H(k+1)x(k+1,k)] \tag{2.2.14}$$

$$x(1,k+1) = x(1,k) + C(1,k)\ dx(k+1) \tag{2.2.15}$$

$$\begin{aligned} x(k+1-L,k+1) = {} & x(k+1-L,k-L) + A(k-L)^{-1}\Delta x(k) \\ & + C(k+1-L,k)\ dx(k+1) \end{aligned} \tag{2.2.16}$$

$$\Delta x(k) = x(k-L,k) - x(k-L,k-L) \tag{2.2.17}$$

The significance of the most important of the above equations may be described as follows:

Equations (2.2.1) and (2.2.12). Covariance and parameter vector propagation from time-point k+1 in the absence of any measurements passed time-point k.

Equations (2.2.6) and (2.2.13). Covariance and parameter vector update at time-point k+1, after the measurements at time-point k+1 have been processed.

Equations (2.2.9) and (2.2.15). [Fixed point smoother] Estimated covariance and vector at the initial time-point after the measurements at the first k+1 time-points have been processed.

Equations (2.2.10) and (2.2.16). [Fixed lag smoother] Estimated covariance and vector L time intervals prior to latest measurement point. As can be seen from equation (2.2.11) and (2.2.17) the smoother estimate for the previous time-point is always needed in the calculations. The fixed point smoother is used to get such an estimate for the initial time-point.

2.3 Change in notation to facilitate programming.

The following definitions are introduced

$$P1(k) = P(k,k) \tag{2.3.1}$$
$$P2(k) = P(k+1,k) \tag{2.3.2}$$
$$PS(k) = P(1,k) \qquad \text{if } k \leq L \tag{2.3.3}$$
$$PS(k) = P(k-L,k) \qquad \text{if } k > L \tag{2.3.4}$$

$$C1(k) = C(h,k), \text{ where} \tag{2.3.5}$$

$$h \equiv 1 \pmod{L} \tag{2.3.6}$$

and

$$k-L < h \leq k \tag{2.3.7}$$

$$C2(k) = C(k+1-L,k) \tag{2.3.8}$$

$$x1(k) = x(k,k) \tag{2.3.9}$$
$$x2(k) = x(k+1,k) \tag{2.3.10}$$
$$xS(k) = x(1,k) \qquad \text{if } k \leq L \tag{2.3.11}$$
$$xS(k) = x(k-L,k) \qquad \text{if } k > L \tag{2.3.12}$$

Equations (2.2.1) through (2.2.17) may now be rewritten in the new notation. Since there is no risk of confusion, the indices for Q, M, G, H, R, dP, ΔP, dx and Δx will be dropped. We find that

$$P2(k) = M\,P1(k)M^T + Q \tag{2.3.13}$$

$$A(k) = P1(k)M^T P2(k)^{-1} \tag{2.3.14}$$

$$C1(k) = A(k) \text{ for } k \equiv 1 \pmod{L} \tag{2.3.15}$$

$$C1(k) = C1(k-1)A(k) \qquad \text{for } k \not\equiv 1 \pmod{L} \tag{2.3.16}$$

$$C2(k) = A(k-L)^{-1}\ C2(k-1)A(k) \quad \text{for } k \not\equiv 0 \ (\text{mod } L) \tag{2.3.17}$$

$$C2(k) = C1(k) \quad \text{for } k \equiv 0 \ (\text{mod } L) \tag{2.3.18}$$

[Note that repeated use of equation (2.3.17) for the computation of C2 will result in numerical inaccuracy. C2 is, therefore, reset every L cycles using equation (2.3.18)]

$$P1(k+1) = P2(k) - dP \tag{2.3.19}$$

$$dP = GH\ P2(k) \tag{2.3.20}$$

$$G = PH^T[R + H\ P2(k)\ H^T]^{-1} \tag{2.3.21}$$

$$PS(k+1) = PS(k) - C1(k)\ dP\ C1(k)^T \quad \text{for } k \le L \tag{2.3.22}$$

$$\begin{aligned} PS(k+1) = {} & P2(k-L) + A(k-L)^{-1}\Delta P\ A(k-L)^{-T} \\ & - C2(k)\ dP\ C2(k)^T \quad \text{for } k > L \end{aligned} \tag{2.3.23}$$

$$\Delta P = PS(k) - P1(k-L) \tag{2.3.24}$$

$$x2(k) = M\, x1(k) \tag{2.3.25}$$

$$x1(k+1) = x2(k) + dx \tag{2.3.26}$$

$$dx = G[Z - H\ x2(k)] \tag{2.3.27}$$

$$xS(k+1) = xS(k) + C1(k)dx, \quad \text{for } k \leq L \tag{2.3.28}$$

$$xS(k+1) = x2(k-L) + A(k-L)^{-1}\Delta x + C2(k)dx \quad \text{for } k > L \tag{2.3.29}$$

$$\Delta x = xS(k) - x1(k-L) \tag{2.3.30}$$

It can be seen from the above that variables P1, P2, A, x1 and x2 require L storage blocks each, but for the remaining quantities only the last computed value need be maintained in computer memory.

2.4 Derivation of equations

The state-transition matrix $M(k+1,k)$ relates the true parameter vector $xT(k)$ at time-point k to the corresponding vector at time-point k+1 through the equation

$$xT(k+1) = M(k+1,k)xT(k) + W(k), \tag{2.4.1}$$

where the term $W(k)$ arises due to our lack of knowledge of the system. $W(k)$ is thus unknown to us. However, we shall assume that

$$EW(k) = 0, \; EW(k)W(k)^T = Q(k) \tag{2.4.2}$$

and $W(k)$ and $W(m)$ are assumed to be uncorrelated if $k \neq m$. For the forward prediction formula we obtain a similar formula

$$x(k+1,k) = M(k+1,k)x(k,k) \tag{2.4.3}$$

[Same as equation 2.2.12.]
whence the error $\tilde{x}$ must satisfy,

$$\tilde{x}(k+1,k) = M(k+1,k)\tilde{x}(k,k) - W(k) \tag{2.4.4}$$

The covariance propagation equation (2.2.1) follows from the above and equation (2.4.2).

The backward prediction formula corresponding to equation (2.4.3) is given by

$$b(k,k+1) = M(k,k+1)b(k+1,k+1) \tag{2.4.5}$$

with the error $\tilde{b}$ being given by

$$\tilde{b}(k,k+1) = M(k,k+1)\,[\tilde{b}(k+1,k+1) + W(k)] \tag{2.4.6}$$

From the above and equation (2.4.2) it follows that the covariance is given by

$$B(k,k+1) = M(k,k+1)[B(k+1,k+1) + Q(k)]M(k,k+1)^T \tag{2.4.7}$$

whence

$$B(k+1,k+1) = M(k+1,k)B(k,k+1)M(k+1,k)^T - Q(k) \tag{2.4.8}$$

Adding equations (2.2.1) and (2.4.8) we obtain

$$P(k+1,k)+B(k+1,k+1) = M(k+1,k)[P(k,k)+B(k,k+1)]M(k+1,k)^T \tag{2.4.9}$$

Since the forward parameter estimate, $x(k,k)$, is based on measurements 1 through k, and the backward estimate, $b(k,k+1)$, is based on measurements k+1 through N, it follows that the two estimates are statistically independent. We may thus use formulae (B.1) and (B.2) in Appendix B to get the smoothed solution at time-point k:

$$x(k,N) = P(k,N)[P(k,k)^{-1}x(k,k)+B(k,k+1)^{-1}b(k,k+1)], \qquad (2.4.10)$$

$$P(k,N) = [P(k,k)^{-1} + B(k,k+1)^{-1}]^{-1} \qquad (2.4.11)$$

The above argument also applies to estimates $x(k,k-1)$ and $b(k,k)$, which may be combined to yield formulae similar to (2.4.10) and (2.4.11). At time-point $k+1$ we thus obtain

$$x(k+1,N) = P(k+1,N)[P(k+1,k)^{-1}x(k+1,k)+B(k+1,k+1)^{-1} \\ b(k+1,k+1)], \qquad (2.4.12)$$

$$P(k+1,N) = [P(k+1,k)^{-1}+B(k+1,k+1)^{-1}]^{-1} \qquad (2.4.13)$$

From the above equation we deduce that

$$P(k+1,N) = B(k+1,k+1)[P(k+1,k) + B(k+1,k+1)]^{-1}P(k+1,k) \\ = P(k+1,k) - P(k+1,k)[P(k+1,k) + B(k+1,k+1)]^{-1}P(k+1,k)$$

From the above and equation (2.4.9) we obtain

$$P(k+1,N) = P(k+1,k) \\ -P(k+1,k)M(k+1,k)^{-T}[P(k,k) + B(k,k+1)]^{-1} \\ M(k+1,k)^{-1}P(k+1,k) \qquad (2.4.14)$$

Since by equation (2.2.2)

$$P(k+1,k)M(k+1,k)^{-T} = A(k)^{-1}P(k,k). \qquad (2.4.15)$$

it follows that

$$P(k+1,N) = P(k+1,k)$$
$$-A(k)^{-1}P(k,k)[P(k,k) + B(k,k+1)]^{-1}P(k,k)A(k)^{-T} \qquad (2.4.16)$$

From equation (2.4.11) we find that

$$P(k,N) = P(k,k)[P(k,k) + B(k,k+1)]^{-1}B(k,k+1)$$
$$= P(k,k) - P(k,k)[P(k,k) + B(k,k+1)]^{-1}P(k,k) \qquad (2.4.17)$$

From the above and equation (2.4.16) we deduce that

$$P(k+1,N) = P(k+1,k) + A(k)^{-1}[P(k,N)-P(k,k)]A(k)^{-T} \qquad (2.4.18)$$

Let

$$F(k) = P(k,N) - P(k,N+1) \qquad (2.4.19)$$

It then follows from equation (2.4.18) that

$$F(k+1) = A(k)^{-1} F(k)A(k)^{-T} \qquad (2.4.20)$$

and hence that

$$F(k) = A(k) F(k+1) A(k)^{T} \qquad (2.4.21)$$

From the above we deduce that

$$F(k) = A(k)A(k+1)\ldots A(N)F(N+1)A(N)^{T}\ldots A(k)^{T} \qquad (2.4.22)$$

For $j \leq k$, Define $C(j,k)$ by

$$C(j,k) = A(j)A(j+1)\ldots A(k) \qquad (2.4.23)$$

It can easily be seen that the above definition is consistent with equations (2.2.3) through (2.2.5). From equations (2.4.22) and (2.4.23) we obtain

$$F(k) = C(k,N)F(N+1)\, C(k,N)^T \tag{2.4.24}$$

It follows from equations (2.4.19) and (2.4.24) that

$$P(k,N+1) = P(k,N)-C(k,N)[P(N+1,N)-P(N+1,N+1)]\, C(k,N)^T \tag{2.4.25}$$

Replacing k by 1 and N by k in the above equation we obtain

$$P(1,k+1) = P(1,k) - C(1,k)dP(k+1)C(1,k)^T \tag{2.4.26}$$

where

$$dP(k+1) = P(k+1,k) - P(k+1,k+1) \tag{2.4.27}$$

The derivation of equations (2.2.6) through (2.2.8) is given in Appendix C. Since equations (2.4.27) and (2.2.6) are equivalent it follows that equations (2.4.26) and (2.2.9) are identical.

In order to derive equation (2.2.10) we first substitute k-L for k and k for N in equation (2.4.18) thus obtaining

$$P(k-1-L,k)=P(k+1-L,k-L)+A(k-L)^{-1}\Delta P(k)A(k-L)^{-T} \tag{2.4.28}$$

where $\Delta P(k)$ is given by equation (2.2.11)

In equation (2.4.25) we substitute k+1-L for k and k for N:

$$P(k+1-L,k+1)=P(k+1-L,k)-C(k+1-L,k)dP(k+1)C(k+1-L,k)^T \quad (2.4.29)$$

where $\delta P(k+1)$ is defined by equation (2.4.27). Equations (2.4.28) and (2.4.29) may be seen to be equivalent to equation (2.2.10). This completes the derivation of equations (2.2.1) through (2.2.12). Equations (2.2.13) and (2.2.14) are derived in Appendix C.

We shall now proceed to derive equations (2.2.15) through (2.2.17). It follows from equations (2.4.12) and (2.4.13) that

$$\begin{aligned} x(k+1,N) &= b(k+1,k+1) \\ &+ P(k+1,N)P(k+1,k)^{-1}[x(k+1,k)-b(k+1,k+1)] \end{aligned} \quad (2.4.30)$$

From the above and equations (2.4.18), (2.4.3) and (2.4.5) we deduce that

$$\begin{aligned} x(k+1,N) &= b(k+1,k+1) + [x(k+1,k)-b(k+1,k+1)] \\ &+A(k)^{-1}[P(k,N)-P(k,k)]A(k)^{-T}P(k+1,k)^{-1} \cdot \\ &\quad M(k+1,k)[x(k,k)-b(k,k+1)] \end{aligned} \quad (2.4.31)$$

whence by equation (2.2.2),

$$\begin{aligned} x(k+1,N) &= x(k+1,k) \\ &+A(k)^{-1}[P(k,N)-P(k,k)]P(k,k)^{-1}[x(k,k)-b(k,k+1)], \end{aligned}$$

i.e.,

$$x(k+1,N) = x(k+1,k)+A(k)^{-1}\{-x(k,k) + P(k,N)\{P(k,k)^{-1}[x(k,k)-b(k,k+1)] + P(k,N)^{-1}b(k,k+1)\}\},$$

whence by equations (2.4.11) and (2.4.10)

$$x(k+1,N) = x(k+1,k)+A(k)^{-1}[x(k,N)-x(k,k)] \tag{2.4.32}$$

Let $f(k) = x(k,N+1) - x(k,N)$ (2.4.33)

Hence $f(k+1) = A(k)^{-1}f(k)$ (2.4.34)

and $f(k) = A(k)\ A(k+1)\ldots A(N)f(N+1)$ (2.4.35)

Using equation (2.4.23) we deduce that

$$f(k) = C(k,N)f(N+1) \tag{2.4.36}$$

Hence,

$$x(k,N+1) = x(k,N) + C(k,N)dx(N+1), \tag{2.4.37}$$

where

$$dx(N+1) = x(N+1,N+1) - x(N+1,N) \tag{2.4.38}$$

If in the above two equations we substitute 1 for k and k for N we obtain equation (2.2.15) and the equivalent of equation (2.2.13). In order to derive equation (2.2.16) we first substitute k-L for k and k for N in equation (2.4.32), thus obtaining

$$x(k+1-L,k) = x(k+1-L,k-L) + A(k+L)^{-1}\Delta x(k), \quad (2.4.39)$$

where $\Delta x(k)$ is given by equation (2.2.17). In equation (2.4.37) we substitute k+1-L for k and k for N:

$$x(k+1-L,k+1) = x(k+1-L,k) + C(k+1-L,k)dx(k+1), \quad (2.4.40)$$

where dx(k+1) is defined consistently with equation (2.2.13). It is easily seen that equations (2.4.39) and (2.4.40) may be combined to yield equation (2.2.16). This completes the derivation of the filter equations.

3. Modelling of the Drag Coefficient

The drag coefficient is modelled as a constant plus a small perturbation xp, which satisfies the differential equation

$$\dot{x}p(t) = -xp(t)/Tp + vp(t), \tag{3.1}$$

where a dot denotes differentiation with respect to t and vp is a random variable with

$$E\ vp(t) = 0, \text{ and} \tag{3.2}$$

$$E\ vp(t1)vp(t2) = \delta(t2 - t1)\ 2Qp/Tp, \tag{3.3}$$

where $\delta(t)$ is the Dirac delta function.

Making the substitution

$$s = t/Tp, \tag{3.4}$$

and defining

$$xp(t) = x'(s), \tag{3.5}$$

where a prime denotes differentiation with respect to s, we deduce that

$$x''(s) + x'(s) = Tp\ vp(t) \tag{3.6}$$

If further we define v(s) = Tp vp(t) then it follows from equations (3.2), (3.3) and equation (D.38) in Appendix D that

$$E\,v(s) = 0, \text{ and} \tag{3.8}$$

$$\begin{aligned} E\,v(sa)v(sb) &= \delta[(sa-sb)Tp]Tp^2\ 2Q/Tp \\ &= 2Q\ \delta(sa-sb) \end{aligned} \tag{3.9}$$

Comparing equations (3.5) through (3.9) with equations (D.1) through (D.3) we deduce from equations (D.7) and (D.10) that

$$xp(t) = gp\ xp(t_0) + \dot{h}p(t), \tag{3.10}$$

where

$$gp = \exp[-(t-t_0)/Tp], \tag{3.11}$$

$$E\dot{h}p(t) = 0, \tag{3.12}$$

and

$$E\,\dot{h}p(t)^2 = Qp[1 - gp^2] \tag{3.13}$$

Equations (3.10) through (3.13) may be seen to correspond to equations (2.4.1) and (2.4.2). In order to interpret Qp and Tp we note that if t is much greater than t_0 then gp is negligibly small. We hence find from equations (3.10) and (3.13) that

$$E\,xp(t)^2 = Qp \tag{3.14}$$

From equation (D.33) we deduce with the aid of equation (3.4) and (3.5) that

$$\frac{E\,xp(t)xp(t+\Delta t)}{E\,xp(t)xp(t)} = \exp(-\Delta t/Tp) \tag{3.15}$$

This concludes the description of the modelling of the drag coefficient.

3.1 Simulation of Drag Coefficients.

Equations (3.10) through (3.13) are used in the simulation of drag coefficients. $\dot{h}p(t)$ is chosen as a normally distributed random variable satisfying equations (3.12) and (3.13).

4. Modelling of Clock Bias

The clock bias $b(t)$ is modeled as the integral of the clock bias rate $\dot{b}(t)$, which itself satisfies the differential equation

$$\ddot{b}(t) = -\dot{b}(t)/Tb + vb(t), \qquad (4.1)$$

where a dot denotes differentiation with respect to t and vb is a random variable with

$$E\ vb(t) = 0, \text{ and} \qquad (4.2)$$

$$E\ vb(t_1)vb(t_2) = \delta(t_2 - t_1)2\ Qb/Tb^2 \qquad (4.3)$$

where $\delta(t)$ is the Dirac delta function.

Making the substitution

$$s = t/Tb \qquad (4.4)$$

defining

$$\dot{b}(t) = x(s) \qquad (4.5)$$

and denoting differentiation with respect to s by a prime, it follows that

$$x''(s) + x'(s) = v(s), \qquad (4.6)$$

where

$$v(s) = vb(t)\ Tb^2 \qquad (4.7)$$

From equations (4.2), (4.3), (4.7) and (D.38) in Appendix D, we conclude that

$$E\ v(s) = 0 \tag{4.8}$$

and

$$E\ v(s_1)v(s_2) = \delta(s_2 - s_1)\ 2\ Qb\ Tb \tag{4.9}$$

Comparing equations (4.6), (4.8) and (4.9) with equations (D.1) through (D.3) in Appendix D, we conclude with the aid of equations (D.6) through (D.12) and (4.4) and (4.5) that

$$b(t) = b(t_0) + (1\text{-}gb)b'(t_0) + h(t), \tag{4.10}$$

$$b'(t) = gb\ b'(t_0) + h'(t), \tag{4.11}$$

where

$$gb = \exp[(t_0\text{-}t)/Tp]. \tag{4.12}$$

and h(t), h'(t) are random variables satisfying

$$E\ h(t) = E\ h'(t) = 0 \tag{4.13}$$

$$E\ h(t)^2 = Qb[2(t\text{-}t_0) - Tb(1\text{-}gb)(3\text{-}gb)] \tag{4.14}$$

$$E\ h'(t)^2 = Qb\ Tb(1\text{-}gb^2) \tag{4.15}$$

$$E\ h(t)h'(t) = Qb\ Tb(1\text{-}gb)^2 \tag{4.16}$$

From equations (4.15), (4.11) and (4.4) we deduce that for large values of t

$$E\,\dot{b}(t)^2 = Qb/Tb \qquad (4.17)$$

From equations (D.33), (D.34), (D.35), (D.3), (4.4), (4.5) and (4.9) we find that for large values of t

$$\frac{E\,\dot{b}(t)\dot{b}(t+\Delta t)}{E\,\dot{b}(t)\dot{b}(t)} = \exp\,(-\Delta t/Tb), \qquad (4.18)$$

$$E[b(t)-b(t+\Delta t)]^2 = 2Q[\Delta t-Tb(1-\exp-\Delta t/Tb)], \qquad (4.19)$$

If Δt is small compared with Tb the above equation reduces to

$$E[b(t) - b(t+\Delta t)]^2 = (Q/Tb)(\Delta t)^2 \qquad (4.20)$$

4.1 Simulation of Clock Bias.

Equations (4.10) through (4.16) are used in the simulation of clock bias. First h(t) is chosen as a normally distributed random number satisfying equations (4.13) and (4.14). Defining a, e, c by

$$a = E\,h(t)^2, \quad e = E\,h'(t)^2, \quad c = E\,h(t)h'(t), \qquad (4.21)$$

h'(t) is then computed as

$$h'(t) = h(t)\,c/a + k(t), \qquad (4.22)$$

where $k(t)$ is another normally distributed random number satisfying

$$E\ k(t) = 0 \tag{4.23}$$

and

$$E\ k(t)^2 = e - c^2/a \tag{4.24}$$

Since $h(t)$ and $k(t)$ are independently chosen random variables, it follows that $h(t)$ and $h'(t)$, computed as described above, will be consistent with equations (4.13) through (4.16).

5. The State Transition Matrix and the Linearization of the Equations of Motion

5.1 Definition of terms and linearization of the equations of motion.

Y	six parameter nominal position-velocity vector	
P	nominal (and constant) drag coefficient	
$Y + y$	perturbed six parameter position-velocity vector	
$P + p$	perturbed drag coefficient	
b	clock bias	
b'	clock bias rate	
x	Nine parameter state vector	
$x^T =$	(y^T, p, b, b')	(5.1.1)
$\dot{Y} =$	$F(Y,P)$, Differential equation satisfied by nominal position-velocity vector	(5.1.2)
$\dot{Y} + \dot{y} =$	$F(Y + y, P + p)$, Differential equation satisfied by perturbed position velocity vector	(5.1.3)
$\dot{Y} + \dot{y} =$	$F(Y,P) + Fy(Y,P)y + Fp(Y,P)p$, Linearized form of equation (5.1.3). $Fy(Y,P)$ is a 6 x 6 matrix of partial derivatives and $Fp(Y,P)$ is a 6-vector of partial derivatives	(5.1.4)
$\phi(t,s) =$	$\partial y(t)/\partial y(s)$, 6 x 6 state-transition matrix	(5.1.5)
$D(t,s) =$	$\partial y(t)/\partial P$ where $D(s,s) = 0$. Vector of partial derivatives of the 6 parameter state vector with respect to a constant change in the drag coefficient at time s.	(5.1.6)
$M(t,s) =$	$\partial x(t)/\partial x(s)$, 9 x 9 state-transition matrix	(5.1.7)
$G(t) =$	$Fy(Y,P)$, $H(t) = Fp(Y,P)$, where Y is a function of time but P is a constant	(5.1.8)
e	Epoch Time (start time of integration of nominal orbit)	

5.2 Computational equations for the state transition matrix.

$$M(t,s) = \begin{bmatrix} \phi(t,s) & D(t,s) & 0 & 0 \\ 0^T & gp & o & o \\ 0^T & o & 1 & 1-gb \\ 0^T & o & o & gb \end{bmatrix} \tag{5.2.1}$$

where 0 is a six dimensional null vector and *o* is a scalar zero.

$$\phi(t,s) = \phi(t,e)\phi(s,e)^{-1} \tag{5.2.2}$$

$$D(t,s) = D(t,e) - \phi(t,s)D(s,e) \tag{5.2.3}$$

5.3 Derivation of equations.

The right-hand side of equation (5.2.1) follows from the definitions of $M(t,s)$, $\phi(t,s)$ and $D(t,s)$ and also from equations (3.10), (4.10) and (4.11). Equations (5.2.2) and (5.2.3) remain to be derived. It follows from equations (5.1.2), (5.1.4) and (5.1.8) that

$$\dot{y}(t) = G(t)y(t) + H(t)p(t). \tag{5.3.1}$$

$\phi(t,e)$ is obtained as the solution of the differential equation

$$\dot{\phi}(t,e) = G(t)\phi(t,e) \quad \text{with } \phi(e,e) = I, \tag{5.3.2}$$

and $D(t,e)$ as the solution of

$$\dot{D}(t,e) = G(t)D(t,e) + H(t), \tag{5.3.3}$$

$$\text{with } D(e,e) = 0 \tag{5.3.4}$$

We wish to show that

$$y(t) = \phi(t,s)y(s) + D(t,s)p(s) \tag{5.3.5}$$

is a solution of equation (5.3.1) and furthermore that in order that the left and right hand sides of equation (5.3.5) be consistent,

$$\phi(s,s) = I \text{ and } D(s,s) = 0 \tag{5.3.6}$$

Differentiating equation (5.3.5) with respect to t we deduce with the aid of equations (5.2.2), (5.2.3), (5.3.2) and (5.3.3) that

$$\dot{y}(t) = G(t)\phi(t,s)y(s) + \left[G(t)D(t,e) + H(t) - G(t)\phi(t,s)D(s,e)\right]p(s),$$

i.e.,

$$\dot{y}(t) = G(t)\phi(t,s)y(s) + \left[G(t)D(t,s) + H(t)\right]p(s)$$

$$= G(t)\left[\phi(t,s)y(s) + D(t,s)p(s)\right] + H(t)p(s).$$

With the aid of equation (5.3.5) the above equation can be seen to reduce to

$$\dot{y}(t) = G(t)y(t) + H(t)p(s) \tag{5.3.7}$$

Equations (5.3.6) can be seen to follow from equations (5.2.2) and (5.2.3). Equation (5.3.7), however, is not identical to equation (5.3.1), but it is a good approximation to it provided that the drag perturbation p changes but little in the time interval (s,t). This we shall assume to be the case.

6. Simulation of Measurements Involving the Global Positioning System (GPS)

6.1 Brief Description of GPS

GPS consists of a set of satellites, whose positions and velocities are known to all users of the system. These satellites transmit radio signals at fixed intervals. The clocks of the GPS satellites are extremely accurate. They are also mutually synchronized. If the user's clock also were synchronized with the GPS clocks, then the user could calculate his distance to each GPS satellite (provided of course that he could see it). Given three distances to three known positions, the user may then solve a simple geometric problem to obtain his own position. If the user clock is not very accurate, then the user may instead process the signal from a fourth satellite to give similar results.

Specifically, GPS consists of 24 satellites arranged in 3 rings of 8 equally spaced satellites (see Figures 6.1 and 6.2). Each satellite is in a 12 hour (26610 km radius) circular orbit with an orbital inclination of 63 degrees. The longitudes of the ascending nodes of the satellite orbits are 0 degrees for those in ring 1, 120 degrees for those in ring 2, and 240 degrees for those in ring 3. Since the satellites of each ring are equally spaced the angular distance between them must be 45 degrees. For each ring a satellite must thus cross the equator from South to North every 90 minutes (another satellite simultaneously crosses from North to South). The satellites of the three rings are phased relative to each other such that a satellite will cross the equator South to North every 30 minutes. The order is ring 1, ring 2, ring 3, ring 1,.... The three rings must obviously intersect one another. However, no two satellites will ever approach each other closer than 10.4 degrees. The orbital paths intersect each other at a latitude of 44.5 degrees (North and South). At the point of intersection, satellites of two different rings approach each other at 101 degrees. The longitudes of the intersections in the Northern Hemisphere occur at 30 degrees (1 ascending, 3 descending), 150 degrees (2 ascending, 1 descending) and 270 degrees (3 ascending, 2 descending).

6.2 *Cartesian Coordinates of the GPS Satellites*

The position of each GPS satellite may be specified (i) by the longitude Ω of the ascending node of its orbit (0 degrees for ring 1, 120 degrees for ring 2, 240 degrees for ring 3), and (ii) by its angular distance ω from that node. ω is computed from the formula

$$\omega = \omega_o + n\Delta\omega + \dot{\omega}t, \qquad (6.2.1)$$

where ω_o equals 0 degrees for ring 1, 30 degrees for ring 2, and 15 degrees for ring 3,

$\Delta\omega$ equals 45 degrees,

n is the satellite number ($n = 0, 1, \ldots, 7$ for each ring),

$\dot{\omega} = 360$ degrees/12 hours, and t is time from midnight.

Let $\bar{p}$ and $\bar{q}$ be two unit vectors lying in the orbital plane, $\bar{p}$ pointing towards the ascending node and $\bar{q}$ pointing towards the point of highest latitude. Then

$$\bar{p} = (\cos \Omega,\ \sin \Omega,\ 0) \quad \text{and} \tag{6.2.2}$$

$$\bar{q} = (-\sin \Omega \cos i,\ \cos \Omega \cos i,\ \sin i), \tag{6.2.3}$$

where i is the orbital inclination. The satellite position $\bar{r}$ is then given by

$$r = \bar{p} \cos \omega + \bar{q} \sin \omega \tag{6.2.4}$$

6.3 GPS Measurements and their Partial Derivatives

The Cartesian coordinates of each GPS satellite is given by a formula of the form (6.2.4). To distinguish between the different satellites we add a subscript. Thus, $\bar{r}_j$, ($j = 1, 2, \ldots 24$) is defined as the position vector of the j-th GPS satellite. Similarly we define $\bar{r}$ as the position vector of the user. The GPS measurement to GPS satellite number j is then given by

$$d_j = \sqrt{(\bar{r}_j - \bar{r})^T (\bar{r}_j - \bar{r})} + b, \tag{6.3.1}$$

where b is a bias term due to a user clock error. We define the unit vector v_j by

$$v_j = (\bar{r}_j - \bar{r})/(d_j - b) \tag{6.3.2}$$

Differentiating d_j with respect to $\bar{r}$ we obtain

$$\partial d_j/\partial \bar{r} = -v_j \tag{6.3.3}$$

Also

$$\partial d_j/\partial b = 1 \tag{6.3.4}$$

The GPS measurement vector (Z(k) in Section 2) is made up of four measurements of the form (6.3.1). The partial derivative matrix (H(k) in Section 2) is made up of the corresponding partial derivatives as given by equations (6.3.3) and (6.3.4).

6.4 GPS Simulations

The problem to be solved in GPS simulations, just as in real situation scheduling, is how to choose 4 GPS satellites out of 24 so as to be able to derive the best possible user position. Since a user satellite has a clear 'horizon', he can see roughly a hemisphere of GPS satellites. This, on the average, amounts to 12 satellites. There are 495 different ways to pick 4 out of 12. To find the best 4 it is necessary to test each combination. To do so at every time point is impractical. The following is a suboptional but good scheme for the selection process. (Further details are given in Appendix E.)

6.4.1. Satellite visibility. It is of course necessary that each selected GPS satellite be visible to the user. In order that the satellite not be visible two criteria must be met

(i) The satellite must appear below the user's 'horizon', i.e.

$$\bar{v}_j^T \bar{r} < 0, \tag{6.4.1}$$

(ii) The satellite to user line of sight must intersect the Earth, defined for this purpose as including an atomsphere 100 km above the surface, i.e.

$$\bar{r}^T\bar{r} < (\bar{v}_j^T\bar{r}) + r_e^2 \tag{6.4.2}$$

where r_e is the radius of the earth as defined above.

6.4.2. Selection of first satellite. The first satellite is somewhat arbitrarily chosen as the one that is highest in the sky ($v_j^T\bar{r}$ is a maximum). There is no loss of generality in designating this as satellite number one (j = 1).

6.4.3 Selection of second satellite. It can be shown (see Appendix E) that the optimal geometric configuration obtains when the angles between the four lines of sight are all equal. This is possible only if the angles equal $\cos^{-1}(-1/3)$ or 109.5 degrees. The second satellite is therefore chosen such that

$$|v_1^T v_j + \frac{1}{3}|$$

is a minimum. This satellite is designated satellite number 2 (j = 2).

6.4.4 Selection of the third satellite. It is shown in Appendix E that given two satellites (1 and 2) then the optimal geometric lines of sight for satellites 3 and 4 must satisfy:

(i) v_3 and v_4 lie in a plane perpendicular to that defined by v_1 and v_2,

(ii) $v_3 + v_4$ is diametrically opposite to $v_1 + v_2$.

(iii) If the angle between v_1 and v_2 is 2α, and the angle between v_3 and v_4 is 2β, then

$$\cos\beta = .327 \cos\alpha - .765 \qquad (6.4.2)$$

The procedure then is to define two unit vectors u_3 and u_4 satisfying the above three criteria and then finding the third satellite such that $v_j^T u$, where $u = u_3$ or u_4, is maximized. This satellite is designated number 3 ($j = 3$). u_3 and u_4 are computed as follows:

$$\cos\alpha = \sqrt{(1 + v_1^T v_2)/2} \qquad (6.4.3)$$

$\cos\beta$ is then computed using equation (6.4.2).

Hence

$$\sin\beta = \sqrt{1 - \cos^2\beta} \qquad (6.4.4)$$

The vector c is defined by

$$c = (v_1 + v_2) \cos\beta / 2\cos\alpha, \qquad (6.4.5)$$

and the vector e by

$$e = (v_1 \wedge v_2) \sin\beta \left[1 - (v_1^T v_2)^2\right]^{-\frac{1}{2}} \qquad (6.4.6)$$

Then

$$u_3 = c + e \quad \text{and } u_4 = c - e \qquad (6.4.7)$$

6.4.5 Selection of the fourth satellite. Given 3 satellites the fourth one is selected optimally as follows: (For derivation of the formulae see Appendix E). Define the matrix

$$A_0 = [v_1, v_2, v_3]^T . \qquad (6.4.8)$$

Then

$$A_0^{-1} = [v_2 \wedge v_3, v_3 \wedge v_1, v_1 \wedge v_2] / [(v_1 \wedge v_2) \cdot v_3] \qquad (6.4.9)$$

Define the vector d_0 by

$$d_0 = [1, 1, 1]^T \qquad (6.4.10)$$

Then compute

$$e_0 = A_0^{-1} d_0, \text{ and } e_{00} = A_0^{-T} e_0 \qquad (6.4.11)$$

The fourth satellite is then found by minimizing the expression

$$2 h_4 e_{00}^T f + h_4^2 e_0^T e_0 (1 + f^T f), \qquad (6.4.12)$$

where

$$h_4 = (1 - v_4^T e_0)^{-1} \qquad (6.4.13)$$

and

$$f = A_0^{-T} v_4 \qquad (6.4.14)$$

FIGURE 6.1

GPS SATELLITE SYSTEM

AS VIEWED FROM A DISTANT POINT AT 30° LATITUDE

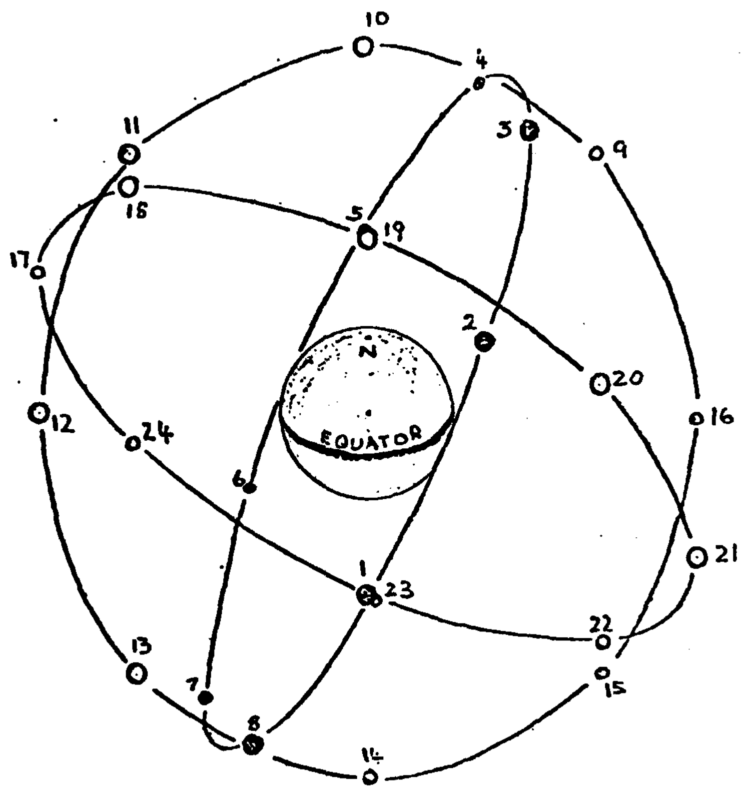
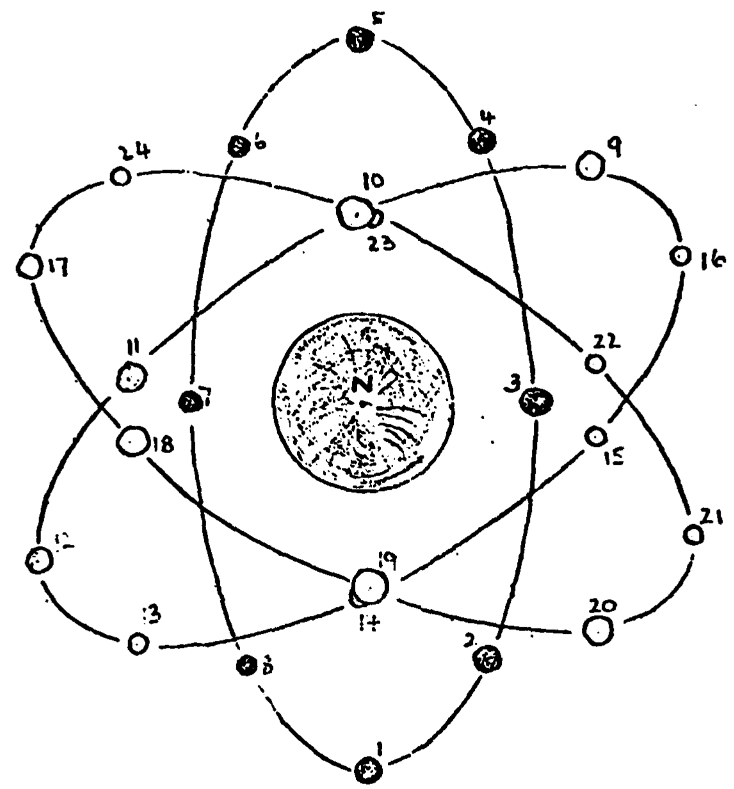


FIGURE 6.2

GPS SATELLITE SYSTEM

AS VIEWED FROM A DISTANT POINT ABOVE THE NORTH POLE



7. Drag Segments in an Orbital Arc.

In most cases the drag coefficient of an orbital satellite is a constant. This is true even if the shape of the satellite is not spherically symmetric, provided that the satellite presents the same aspect angle along its direction of motion. In the equations of motion the drag coefficient always appears as a factor multiplying the atmospheric density. It is thus possible to compensate for density variations through corresponding changes in the drag coefficient. This is often done in practice. The capability to do that has now been added to Photonap. The program has been modified to include a number of different drag segments. Each drag coefficient may be constrained either to some a priori value (absolute constraint) or the coefficients of contiguous segments may be constrained relative to each other (relative constraints).

7.1 Partial Derivatives of the State Vector with Respect to the Drag Coefficients.

Let

$$\dot{Y} = F(Y, P_k), \tag{7.1}$$

denote the differential equation governing the satellite motion. Y is the state vector (position-velocity vector) and P_k is the drag coefficient in segment k valid in the interval between times t_k and t_{k+1}. Let y and p_k denote a small perturbations in Y and P_k, respectively. Then

$$\dot{y} = \frac{\partial}{\partial Y} F(Y,P_k)\ y + \frac{\partial}{\partial P_k} F(Y,P_k) p_k \tag{7.2}$$

Writing $$G = \partial F(Y,P_k)/\partial Y \tag{7.3}$$

and

$$H = \partial F(Y,P_k)/\partial P_k \tag{7.3}$$

the above equation may be written as

$$\dot{y} = Gy + Hp_k \tag{7.4}$$

To indicate that y, G and H are functions of time we rewrite the equation in the form

$$\dot{y}(t) = G(t)y(t) + H(t)p_k \tag{7.5}$$

The 6 x 6 state transition matrix $\phi(t,e)$ is defined as the solution of the differential equation

$$\dot{\phi}(t,e) = G(t)\phi(t,e), \tag{7.6}$$

with

$$\phi(e,e) = I, \tag{7.7}$$

e being the time of Epoch.

The six-vector of drag partials, $D_k(t)$ is defined as the solution of the differential equation

$$\dot{D}_k(t) = G(t)D_k(t) + H(t), \tag{7.8}$$

with $D_k(t_k) = 0,$ (7.9)

and $D_k(t)$ being defined only in the interval $[t_k, t_{k+1}]$. Equation (7.5) is similarly, for each value of k, valid only in the interval $[t_k, t_{k+1}]$. The solutions of equations (7.5) must be continuous and satisfy the initial conditions

$$y(t_1) = 0, \tag{7.10}$$

where $t_1 = e$, the Epoch Time.

It will now be shown that if

$$t_k \leq t \leq t_{k+1} \tag{7.11}$$

then equation (7.5) is satisfied by

$$x_k(t) = \phi(t,e) \sum_{i=2}^{k} \phi(t_i,e)^{-1} D_{i-1}(t_i)\, p_{i-1}$$

$$+ D_k(t) p_k \tag{7.12}$$

Differentiating equation (7.12) with respect to t, it can easily be seen that equation (7.5) is satisfied. It remains to be shown that the solution is continuous, i.e., that $x_k(t_k) = x_{k-1}(t_k)$. From equation (7.12) we obtain

$$x_{k-1}(t_k) = \phi(t_k,e) \sum_{i=2}^{k-1} \phi(t_i,e)^{-1} D_{i-1}(t_i) p_{i-1}$$

$$+ D_{k-1}(t_k) p_{k-1}, \tag{7.13}$$

The above equation may be rewritten as

$$x_{k-1}(t_k) = \phi(t_k,e) \sum_{i=2}^{k} \phi(t_i,e)^{-1} D_{i-1}(t_i) p_{i-1} \tag{7.14}$$

From equations (7.9), (7.12) and (7.14) we deduce that

$$x_{k-1}(t_k) = x_k(t_k) \tag{7.15}$$

Since $x_1(t_1) = 0$ it follows that $x_k(t)$ as given by equation (7.12) is the required solution of equation (7.5), i.e., for t satisfying equation (7.11),

$$y(t) = \phi(t,e) \sum_{i=2}^{k} \phi(t_i,e)^{-1} D_{i-1}(t_i)p_{i-1} + D_k(t)p_k \tag{7.16}$$

It hence follows that the required partial derivatives are given by

$$\left.\begin{array}{ll} \text{for } t \geq t_{i+1}, & \partial y(t)/\partial p_i = \phi(t,e)\phi(t_{i+1},e)^{-1}D_i(t_{i+1}) \\ \text{for } t_i \leq t \leq t_{i+1}, & \partial y(t)/\partial p_i = D_i(t) \\ \text{for } t \leq t_i, & \partial y(t)/\partial p_i = 0 \end{array}\right\} \tag{7.17}$$

7.2 Absolute and Relative Constraints

To obtain the solution δp of a linearized weighted least squares problem, an equation of the following form must be solved

$$N\ \delta p = b \tag{7.18}$$

In the above equation, N is a positive-definite matrix (the normal equations coefficient matrix) and b a vector (the normal equations vector). Using the summation convention equation (7.18) may be written in index form as

$$N(i,j)\delta p(j) = b(i) \tag{7.19}$$

If all measurements are statistically independent, then N(i,j) and b(i) are computed as the sum of terms of the form

$$\Delta N(i,j) = W \frac{\partial m}{\partial p(i)} \frac{\partial m}{\partial p(j)} \tag{7.20}$$

and

$$\Delta b(i) = W \frac{\partial m}{\partial p(i)} (m_0 - m), \tag{7.21}$$

where W, the measurement weight is inversely proportional to the variance of the measurement error,

m_0 is the observed measurement

and m is the measurement calculated as a function of a set of parameters p(k).

After the solution of equation (7.19) has been obtained the estimated parameter is updated to $p(k) + \delta p(k)$.

7.2.1 Absolute Constraints. If, a priori, we know that parameter $p(i) = a_i$ and that the error in a_i has a variance of σ^2, then we may treat that information in exactly the same way as measurement information. Hence $m_0 = a_i$, $m = p(i)$ and $\partial m/\partial p(i) = 1$.

In accordance with equations (7.20) and (7.21) the contributions to N and b are then given by

$$\Delta N(i,i) = \sigma^{-2} \tag{7.22}$$

$$\Delta b(i) = \sigma^{-2}(a_i - p(i)) \tag{7.23}$$

7.2.2 Relative Constraints. If, a priori, we know that parameters i and j should assume the same value and that the error in this assumption has a variance of σ^2, then we may treat this information exactly the same way as measurement information. Hence $m_0 = 0$, $m = p(i) - p(j)$, $\partial m/\partial p(i) = 1$ and $\partial m/\partial p(j) = -1$.

In accordance with equations (7.20) and (7.21) the contributions to N and b are then given by

$$\left.\begin{aligned} \Delta N(i,i) &= \sigma^{-2} \\ \Delta N(i,j) &= -\sigma^{-2} \\ \Delta N(j,j) &= \sigma^{-2} \end{aligned}\right\} \tag{7.24}$$

and

$$\left.\begin{aligned} \Delta b(i) &= \sigma^{-2}\ p(j) - p(i) \\ \Delta b(j) &= \sigma^{-2}\ p(i) - p(j) \end{aligned}\right\} \tag{7.25}$$

Note that m could equally well have been defined by $m = p(j) - p(i)$. The result, however, would be the same.

8. The Lockheed-Jacchia Atmospheric Model

The equations presented in this section are based partly on some equations supplied by Mr. George Stentz of DMAAC, St. Louis, MO and partly on a computer program listing from the same source. Part of the description comes from (Jacchia, 1960). A modified version of the DMAAC program has been incorporated in Photonap.

8.1 Description of Variables and Constants used in the Program.

Ψ — Angle between point of interest and point of maximum solar heating effect as seen from the center of the Earth.

g — $= \left(\frac{1+\cos\Psi}{2}\right)^3$

λ — = .55 radians. Lag angle. Angle between the sun and the point of maximum solar heating as seen from the center of the Earth.

t — Time (in days) from noon on January 1, 4713 B.C.

J_{DO} — = 2436204. Number of days between January 1, 1958 and January 1, 4713 B.C.

t' — Time (in days) since noon on December 31, 1957.

ω — = .017203 radians/day. The Earth's orbital rate about the sun

L — Longitude of the Sun measured along the ecliptic from the equinox

L_0 — -Longitude of the Sun when $t' = 0$

e — = 0.01675. The eccentricity of the Earth's orbit

ϵ — = .4092 radians. Obliquity of the ecliptic

$F_{10.7}$		10.7 cm flux mea ured in units of 100×10^{-22} watt/m^2/cycle/sec
F_{20}		20 cm flux measured in same units as $F_{10.7}$
C_8	$= 1.5$	constants used in computation of $F_{10.7}$
C_9	$= 0.8$	
ω_F	$= \frac{2\pi}{4020}$	Frequency corresponding to a period of 4020 days (approximately 11 years)
C_{15}	$= 0.85$	Conversion factor for converting $F_{10.7}$ to the equivalent F_{20}
ρ		Atmospheric Density (slugs/cu.ft)
ρ'		$= \frac{1}{\rho} \frac{d\rho}{dh} = \frac{d}{dh} \log \rho$ (1/n.m.)
h		Height above the surface
h_1	$= 76$ n.m.	
h_2	$= 108$ n.m.	
h_3	$= 378$ n.m.	
h_4	$= 1000$ n.m.	
C_{12}	$= 5.606 \times 10^{-12}$ slugs/cu.ft.	
d_1	$= 7.18$ unless otherwise specified by user	
C_{18}	$= 153$ n.m.	
d_2	$= -15.738$ unless otherwise specified by the user	
C_{26}	$= .00368$ (n.m.)$^{-1}$	
C_{27}	$= 6.363$	
C_{28}	$= .0048$ (n.m.)$^{-1}$	
C_{29}	$= 0.19$	
C_{30}	$= .0102$ (n.m.)$^{-1}$	
C_{31}	$= 1.9$	
C_{36}	$= .00504$ (slugs/cu.ft.)(n.m.)^5	
C_{37}	$= 6 \times 10^6$ (n.m.)^3	
1 n.m.	$= 1.852$ km	

1 slug/cu.ft. $= 0.515378 \times 10^{12}$ kg/km^3

8.2 Description of Equations

The Jacchia Atmospheric Model is a dynamic model in the sense that it is a function, not only of position, but also of time. The time dependence is due to solar heating. However, since solar heating is not instantaneous, the maximum perturbation to the atmosphere will occur some time after noon, local time (according to this model just after 2 p.m. local time). If $\bar{s}$ is a unit vector pointing towards the sun, then the maximum perturbation will occur in the direction of $\bar{s}'$, where $\bar{s}$ and $\bar{s}'$ point towards the same latitude, but $\bar{s}'$ towards a point λ radians further East. If $\bar{u}$ is a unit vector pointing towards the point of interest, then $\cos\psi$ is defined by

$$\cos\psi = \bar{u} \cdot \bar{s}' \tag{8.1}$$

If

$$\bar{u} = (x,y,z) \tag{8.2}$$

and

$$\bar{s} = (s_1, s_2, s_3), \tag{8.3}$$

then

$$\bar{s}' = (s_1\cos\lambda - s_2\sin\lambda,\ s_2\cos\lambda + s_1\sin\lambda, s_3) \tag{8.4}$$

and

$$\cos\psi = (s_1x + s_2y)\cos\lambda - (s_2x - s_1y)\sin\lambda + s_3 \tag{8.5}$$

$\bar{s}$ is computed using the following equations

$$t' = t - J_{DO} \tag{8.6}$$

$$L = \omega t' + 2e \sin \omega t' - L_0 \tag{8.7}$$

$$\bar{s} = (\cos L,\ \sin L \cos\varepsilon,\ \sin L \sin\varepsilon) \tag{8.8}$$

Combining equations (8.3), (8.5) and (8.8) we obtain

$$\cos\psi = (x\cos L + y\sin L\ \cos\varepsilon)\cos\lambda$$
$$+ (-x\sin L\ \cos\varepsilon + y\cos L)\sin\lambda + \sin L\sin\varepsilon \qquad (8.9)$$

Let g be defined by

$$g = [\cos\psi/2]^6,$$

i.e.,

$$g = \left(\frac{1+\cos\psi}{2}\right)^3 \qquad (8.10)$$

Unless input by the user the 10.7 cm flux is given by

$$F_{10.7} = C_8 + C_9 + \cos(\omega_F t') \qquad (8.11)$$

This is converted to an equivalent flux at 20 cm by the formulae

$$F_{20} = C_{15}\ F_{10.7} \qquad (8.12)$$

For the purpose of calculating the density, the atmosphere is subdivided into four different regions, with different sets of formulae being valid in each region. These are given below.

<u>Region A</u> $\quad h_1 \leq h \leq h_2$

$$\rho = \rho_1\rho_2\rho_3, \qquad (8.13A)$$

where

$$\rho_1 = c_{12}\left(\frac{h_1}{h}\right)^{d_1} \qquad (8.14A)$$

$$\rho_2 = \left[\frac{h_2-h}{h_2-h_1} + \left(\frac{h-h_1}{h_2-h_1}\right)^{\frac{4}{3}}\ F_{20}\right] \qquad (8.15A)$$

$$\rho_3 = 1 + \frac{h-h_1}{c_{18}} \; g \tag{8.16A}$$

$$\rho' = -\frac{d_1}{h} + \frac{1}{\rho_2} - \left[\frac{1}{h_2-h_1} + \frac{\frac{4}{3}F_{20}}{h_2-h_1}\right] + \frac{g}{c_{18}\rho_3} \tag{8.17A}$$

<u>Region B</u> $h_2 \leq h \leq h_3$

$$\rho = \rho_0 \; q \tag{8.13B}$$

where

$$\rho_0 = 10^{d_2} - c_{26}h + c_{27} \exp(-c_{28}h) \tag{8.14B}$$

and

$$q = F_{20}\left\{1 + c_{29}\left[\exp(c_{30}h) - c_{31}\right]g\right\} \tag{8.15B}$$

$$\rho' = \log_e 10\left[-c_{26} - c_{28}c_{27}\exp(-c_{28}h)\right] + \frac{F_{20}c_{29}c_{30}\exp(c_{30}h)g}{q}$$

<u>Region C</u> $h_3 \leq h \leq h_4$

$$\rho = b_1 b_2, \tag{8.13C}$$

where

$$b_1 = c_{36}\frac{F_{10.7}}{h^5} \tag{8.14C}$$

$$b_2 = g\left(1 - \frac{c_{37}}{h^3}\right) + \frac{c_{37}}{h^3} \tag{8.15C}$$

$$\rho' = -\frac{5}{h} - \frac{3}{h}(1-g)\frac{c_{37}}{h^3} \; b_2 \tag{8.16C}$$

<u>Region D</u> $h \geq h_4$ or $h \leq h_1$

$$\rho = \rho' = 0 \tag{8.13D}$$

Note that in all of the above equations the density is computed in slugs/cu.ft. and ρ' in 1/n.m. Before being used by Photonap these quantities are converted to kg/km^3 and km^{-1}, respectively.

9. An Outline of Program Changes

The program changes made to Photonap fall into two categories: (i) changes to existing routines, and (ii) the addition of new routines. Two of the existing routines, SPOLCD and SOLVER, were initially simply modified, but owing to the routines in the process becoming extremely lengthy and unmanageable, they were later split into smaller routines. Thus SPOLCD was split into SPOLCD, SPOOL1, SPOOL2 and SPO100. SOLVER was split into SOLVER, SOLV1, SOLV2 and SOLV3. The following totally new routines have been added:

A. Routines associated with the Lockheed-Jacchia atmospheric model: JACHIA

B. Routines associated with drag segmentation: DRAGU

C. Routines associated with Kalman filtering and smoothing: KMNCON, KMNEVA, KMNIDE, KMNINI, KMNINV, KMNMP1, KMNMP2, KMNMP3, KMNRAN, KMNOUT, KMNSIM, KMNSM2, INVSYM, INVSYS, MAT99, VXPROD.

D. Routines associated with GPS measurements: SELECT

E. Routines associated with normal equations for correlated measurements: SOLVFU, SOLAWA

In addition to the changes described above, subroutine SVARED, after a trivial change in the coding, was found to be superfluous, and was hence removed from Photonap.

A flow chart of Photonap together with a short description of each routine is given in Appendix F.

10. Changes to the Photonap User's Guide

The following has been added to the User's Guide.

(i) Insertion into Section I of a general description of control card set-ups for Kalman filtering and smoothing.

(ii) Addition to 101 card input to specify Kalman filter mode.

(iii) Description of 230 card for specifying drag segments.

(iv) Addition of note (Note 13) to 601 card for handling of drag coefficients appearing in different drag segments of the same arc.

(v) Description of 612 card for specifying constants required by Lockheed-Jacchia atmosphere

(vi) Description of 614 card for specifying GPS filter constants

(vii) Addition of note (Note 5) to 701 card for processing of correlated measurements output from Kalman filter or smoother.

(viii) Addition to Appendix IB describing tape format for Kalman filter input, and tape format for Kalman filter output.

(ix) Addition of Appendix V describing example of the job control language required for running Photonap on the CDC 6400.

11. Test Runs

In order to check out the modified version of Photonap a large number of test runs were made. These included running the standard set of Photonap test decks, which are run after every program modification. Five new test decks, designated PB4, PB5, PB6, PB7 and PB8, have been added to the standard set, which now consists of

> non-photogrammetric test decks TESTXX, PA1, PA2, PA3, PA4, PA5, PA5X1, PA5X2, PA6, PA7, PA8, PA9, PB0, PB1, PB2, PB3, PB4, PB5, PB6, PB7, PB8,
>
> Photogrammetric test decks PAA, PAB, PAC, PAD, PAE, PAF, PAG, PAGX1,
>
> combined test deck FATBOY.

A short description of each of the new test decks is given below.

11.1 Test Deck PB4. Lockheed-Jacchia atmosphere and multiple drag segments. Two parts.

- (i) Data generation using Jacchia Atmosphere and a single drag segment.
- (ii) Orbit and drag coefficient recovery using U.S. Standard Atmosphere. Six drag segments with relative constraints. Epoch coincident with start of first drag segment.

11.2 Test Deck PB5. Lockheed-Jacchia Atmosphere and multiple drag segments. Two parts.

(i) Data generation using Jacchia Atmosphere and a single drag segment.

(ii) Orbit and drag coefficient recovery using U.S. Standard Atmosphere. Six drag segments with relative and absolute constraints. Epoch in middle of fourth segment.

11.3 Test Deck PB6. Six point smoother using GPS measurements. Three part run.

(i) GPS data generation using Lockheed-Jacchia Atmosphere.

(ii) Six point smoother using U.S. Standard Atmosphere. Smoother output of position and velocity at 30 second intervals.

(iii) Orbit comparison between the smoother output and the orbit used in data generation.

11.4 Test Deck PB7. Filter using GPS measurements. Three part run.

(i) GPS data generation using Lockheed-Jacchia Atmosphere.

(ii) Filter (0 point smoother) using U.S. Standard Atmosphere. Filter output of position and velocity at 30 second intervals.

(iii) Orbit recovery based on Lockheed-Jacchia Atmosphere. Filter output used as measurement data.

11.5 <u>Test Deck PB8</u>. Filter and smoother comparisons.

U.S. Standard Atmosphere used in all three parts:

(i) GPS data generation,

(iia) Filter,

(iib) 16 point smoother.

Comparison between orbit used in generation (Part (i), pages 7 through 9) and the orbit recovered (Parts (ii), pages 4 through 6) shows the superiority of the 16 point smoother over the filter.

12. References

Ballew (1977) — Dr. Robert Ballew, Private communication (oral), May 13, 1977.

Gelb (1977) — Applied Optimal Estimation, Arthur Gelb, (Editor), The M.I.T. Press, Third Printing, February 1977.

Hartwell (1975) — A Photogrammetric and Tracking Network Analysis Program for the UNIVAC 1108 Computer, J. G. Hartwell, J. J. Lynn, and G. E. Morduch, ETL-0018, Contract DACA 76-74-C-0001, May, 1975.

Jacchia (1960) — A Variable Atmospheric-Density Model from Satellite Accelerations. Luigi G. Jacchia, Journal of Geophysical Research, Vol. 65, No. 9, September 1960.

Meditch (1969) — Stochastic Optimal Linear Estimation and Control, J.S. Meditch, McGraw-Hill, 1969.

Morduch (1976) — Use of the GPS Satellite System for the Determination of the Magsat Position, Georg E. Morduch, Contract NAS5-23587, September, 1976.

Stentz (1978) — Mr. George Stentz, Private communication (letter), January 20, 1978.

APPENDIX A

Some formulae for the differentiation of the trace of matrix products.

Given a positive-definite matrix D and general matrices A and F, the following three formulae will be derived.

$$\frac{\partial}{\partial F} \text{Trace } (DFAF^T) = DF(A + A^T) \tag{A.1}$$

$$\frac{\partial}{\partial F} \text{Trace } (DFA) = DA^T \tag{A.2}$$

$$\frac{\partial}{\partial F} \text{Trace } (DAF^T) = DA \tag{A.3}$$

Proof. Denoting the left hand side of formula (A.1) by L, we may express it in index form as

$$L_{ij} = \frac{\partial}{\partial F_{ij}} \sum_{a,b,c,d} D_{ab}F_{bc}A_{cd}F_{ad}$$

$$= \sum_{a,d} D_{ai}A_{jd}F_{ad} + \sum_{b,c} D_{ib}F_{bc}A_{cj}$$

$$= (D^TFA^T + DFA)_{ij}.$$

Since D is symmetric, formula (A.1) follows. Denoting the left hand side of the second formula by M, we similarly obtain:

$$M_{ij} = \frac{\partial}{\partial F_{ij}} \sum_{a,b,c} D_{ab}F_{bc}A_{ca}$$

$$= \sum_{a} D_{ai}A_{ja}$$

$$= (D^T A^T)_{ij}$$

Since D is symmetric, the second formula follows. Denoting the left hand side of the third formula by N, we find that

$$N_{ij} = \frac{\partial}{\partial F_{ij}} \sum_{a,b,c} D_{ab}A_{bc}F_{ac}$$

$$= \sum_{b} D_{ib}A_{bj}$$

$$= (DA)_{ij},$$

which is equivalent to the third formula.

APPENDIX B

The combination of two independent unbiased minimum variance estimates.

Given

(i) two independent unbiased estimates a and b of a parameter vector, whose true value is xT,

(ii) A and B, the covariances of the errors in a and b, respectively,

then the two solutions may be combined to give a new minimum variance solution x, with an associated covariance P, where

$$x = P(A^{-1}a + B^{-1}b) \tag{B.1}$$

and

$$P = (A^{-1} + B^{-1})^{-1} \tag{B.2}$$

Proof. Assuming x to be a linear combination of a and b, we may write it in the form

$$x = Fa + F^1 b \tag{B.3}$$

where the matrices F and F^1 have to be determined. In order that x be unbiased, we must clearly have

$$F^1 = I - F, \tag{B.4}$$

where I is the identity matrix. Denoting the errors in x, a and b by $\tilde{x}$, $\tilde{a}$ and $\tilde{b}$, respectively, we deduce from equations (B.3) and (B.4) that

$$\tilde{x} = F\tilde{a} + (I-F)\tilde{b} \tag{B.5}$$

Since a and b are independent it follows from the above and the definition of the covariance that

$$P = FAF^T + (I-F)B(I-F)^T \tag{B.6}$$

F is chosen such that the expected value of $\tilde{x}^T D\tilde{x}$, where D is a positive-definite matrix, is minimized. It turns out that as long as D is symmetric and non-singular the solution is independent of the choice of D. Remembering that if XY and YX are both square matrices, then trace (XY) = trace (YX), it follows that the quantity we are trying to minimize is the expected value of trace $(D\tilde{x}\tilde{x}^T)$, i.e., trace (DP). From equation (B.6) and formulae (A.1), (A.2) and (A.3) in Appendix A, we then deduce that

$$D[2F(A + B) - 2B] = 0. \tag{B.7}$$

Hence

$$F(A + B) - B = 0, \tag{B.8}$$

and

$$F = (A^{-1} + B^{-1})^{-1}A^{-1} \tag{B.9}$$

From the above and equation (B.4) we find that

$$F^1 = (A^{-1} + B^{-1})^{-1}B^{-1}, \tag{B.10}$$

and from equations (B.6) and (B.8) we obtain,

$$P = B \quad FB$$

$$= BF^1$$

From the above and equation (B.10) it follows that

$$P = (A^{-1} + B^{-1})^{-1},$$

which is equation (B.2). Equation (B.1) follows from the above and equations (B.3), (B.9) and (B.10).

APPENDIX C

Updating the minimum variance estimate based on new independent measurements.

Given

(i) an unbiased estimate a of a parameter vector, whose true value is xT,

(ii) A, the covariance of the error in a,

(iii) a measurement vector Z satisfying the equation

$$Z = H\,xT + r, \text{ where} \tag{C.1}$$

$$E(r) = 0 \qquad E(rr^T) = R, \tag{C.2}$$

and H is a given matrix,

then the new minimum variance estimate x is given by

$$x = a + G(Z-Ha), \tag{C.3}$$

where

$$G = AH^T(R + HAH^T)^{-1}, \tag{C.4}$$

and P, the covariance of the error in x, is given by

$$P = A - GHA \tag{C.5}$$

Proof. Equation (C.3) is clearly a general form for a linear unbiased estimate of x. We shall, therefore, assume that x is given by equation (C.3) and then proceed to derive equations (C.4) and (C.5). Let $\tilde{x}$ and $\tilde{a}$ denote the errors in x and a, respectively. It then follows from equations (C.1), (C.2) and (C.3) that

$$\tilde{x} = \tilde{a} + G(r - H\tilde{a}), \text{ i.e.}$$

$$\tilde{x} = (I - GH)\tilde{a} + Gr \tag{C.6}$$

Since $\tilde{a}$ and $\tilde{r}$, by assumption, are independent, it follows that

$$P = (I-GH)A(I-GH)^T + GRG^T \tag{C.7}$$

G is chosen such that the expected value of $\tilde{x}^T D\tilde{x}$, where D is a positive-definite matrix, is minimized. It turns out that as long as D is symmetric and non-singular the solution is independent of the choice of D. Remembering that if XY and YX are both square matrices, then trace (XY) = trace (YX), it follows that the quantity we are trying to minimize is the expected value of trace $(D\tilde{x}\tilde{x}^T)$, i.e., trace (DP). From equation (C.7) and formulae (A.1), (A.2) and (A.3) in Appendix A, we then deduce that

$$D[2G(HAH^T+R) - 2AH^T] = 0$$

Hence,

$$G(HAH^T+R) - AH^T = 0 \tag{C.8}$$

from which equation (C.4) immediately follows. Post-multiplying equation (C.8) by G^T and subtracting the result of equation (C.7) yields equation (C.5). This completes the proof.

APPENDIX D

Differential equation associated with timing bias and variations in the drag coefficient.

Given the differential equation

$$\ddot{x}(t) + \dot{x}(t) = v(t), \tag{D.1}$$

where a dot denotes differentiation with respect to t, and v(t) is a random variable with

$$Ev(t) = 0 \tag{D.2}$$

$$Ev(ta)v(tb) = V\,\delta(ta - tb), \tag{D.3}$$

$\delta(t)$ being the Dirac delta function satisfying

$$\delta(t) = 0 \quad \text{if } t \neq 0 \text{ and} \tag{D.4}$$

$$\int_{-\infty}^{\infty} \delta(t)dt = 1, \tag{D.5}$$

it will be shown that

$$x(t) = x_0 + \dot{x}_0\,(1\text{-}\exp\text{-}t) + h(t) \tag{D.6}$$

and

$$\dot{x}(t) = \dot{x}_0 \exp(-t) + \dot{h}(t) \tag{D.7}$$

where h(t) and $\dot{h}(t)$ are random variables satisfying

$$Eh(t) = E(\dot{h}(t) = 0, \tag{D.8}$$

$$Eh(t)^2 = \tfrac{1}{2}V[2t - (1\text{-}\exp\text{-}t)(3\text{-}\exp\text{-}t)] \tag{D.9}$$

$$E\dot{h}(t)^2 = \tfrac{1}{2}V[1 - \exp\text{-}2t] \tag{D.10}$$

$$Eh(t)\dot{h}(t) = \tfrac{1}{2}V[1 - \exp\text{-}t]^2 \tag{D.11}$$

and

$$x_0 = x(0), \; \dot{x}_0 = \dot{x}(0) \tag{D.12}$$

<u>Proof</u>. Since

$$\frac{d}{dt}\left[\dot{x}(t)\exp t\right] = \left[\ddot{x} + \dot{x}\right]\exp t, \text{ equation (D.1)}$$

may be integrated to give

$$\dot{x}(t)\exp t - \dot{x}_0 = \int_0^t \exp s \; v(s)ds \tag{D.13}$$

Hence

$$\dot{x}(t) = \dot{x}_0\exp(-t) + \dot{h}(t), \tag{D.14}$$

where

$$\dot{h}(t) = \int_0^t \exp(s-t)v(s)ds \tag{D.15}$$

Since

$$\frac{d}{dt} \int_0^t v(s)\,[1-\exp(s-t)]ds = \int_0^t \exp(s-t)v(s)ds,$$

it follows from equation (D.15) that

$$h(t) = \int_0^t v(s)\,[1-\exp(s-t)]ds \tag{D.16}$$

Integration of equation (D.14) yields

$$x(t) = x_0 + \dot{x}_0\,(1-\exp -t) + h(t) \tag{D.17}$$

Equations (D.6) and (D.7) have thus been derived. Equations (D.8) easily follow from equations (D.15), (D.16) and (D.2). Equations (D.9) through (D.11) will now be derived. From equations (D.15), (D.16) and (D.3) it follows that

$$E\,h(t)^2 = V \int_0^t [1-\exp(s-t)]^2 ds, \tag{D.18}$$

$$E\,\dot{h}(t)^2 = V \int_0^t [\exp 2(s-t)]\,ds, \tag{D.19}$$

$$E\,h(t)\dot{h}(t) = V \int_0^t \exp(s-t)\,[1-\exp(s-t)]\,ds. \tag{D.20}$$

Integrating equation (D.18) we obtain

$$E\,h(t)^2 = V[t - 2(1-\exp-t) + \tfrac{1}{2}(1-\exp-2t)],$$

which after some simplification leads to equation (D.9). Equations (D.19) and (D.10) are easily seen to be equivalent. From equation (D.20) we deduce that

$$E\,h(t)\dot{h}(t) = V[(1-\exp-t) - \tfrac{1}{2}(1-\exp-2t)],$$

which can be seen to reduce to equation (D.11). This completes the derivation of the required equations.

Expected values of x, $\dot{x}$, x^2 and $\dot{x}^2$ for large values of t.

It follows from equations (D.6) through (D.11) that for large values of t,

$$Ex(t) = x_0 + \dot{x}_0 \tag{D.21}$$

$$E\dot{x}(t) = 0 \tag{D.22}$$

$$Ex(t)^2 = (x_0 + \dot{x}_0)^2 + Vt \simeq Vt \tag{D.23}$$

$$E\dot{x}(t)^2 = \tfrac{1}{2}V \tag{D.24}$$

Autocorrelation functions for x and $\dot{x}$.

If ta $\geq$ t then we obtain similarly to equations (D.18) and (D.19)

$$E\,h(t)h(ta) = V \int_0^t [1-\exp(s-t)]\,[1-\exp(s-ta)]\,ds, \tag{D.25}$$

and

$$E\,\dot{h}(t)\dot{h}(ta) = V \int_0^t \exp(s-t)\exp(s-ta)\,ds. \tag{D.26}$$

Hence,

$$E\,h(t)h(ta)/V = t-[1-\exp-t]-[\exp(t-ta)-\exp-ta]$$
$$+\tfrac{1}{2}[\exp(t-ta)-\exp(-t-ta)], \tag{D.27}$$

and

$$E\,\dot{h}(t)\dot{h}(ta)/V = \tfrac{1}{2}[\exp(t-ta)-\exp(-t-ta)] \tag{D.28}$$

Writing $ta = t + \Delta t$, (D.29)

we deduce that for large t,

$$E\, h(t)h(t + \Delta t) = Vt \tag{D.30}$$

and

$$E\, \dot{h}(t)\dot{h}(t + \Delta t) = \tfrac{1}{2}V \exp(-\Delta t) \tag{D.31}$$

Thus for large t,

$$[Ex(t)x(t + \Delta t)]/[Ex(t)^2] = 1 \tag{D.32}$$

and

$$[E\dot{x}(t)\dot{x}(t + \Delta t)]/[E\dot{x}(t)^2] = \exp(-\Delta t) \tag{D.33}$$

A further quantity of interest is $E[x(t) - x(t + \Delta t)]^2$. It follows from equations (D.6) and (D.8) that for large t,

$$x(t) - x(t + \Delta t) = h(t) - h(t + \Delta t).$$

Hence we obtain with the aid of equations (D.9), (D.27) and (D.29)

$$\begin{aligned} E[x(t)-x(t + \Delta t)]^2 &= E\,h(t)^2 + h(t + \Delta t)^2 \\ &\quad - 2\,E\,h(t)h(t + \Delta t) \\ &= V[t-1.5] + V[t + \Delta t - 1.5] \\ &\quad - 2V[t - 1 - \tfrac{1}{2} \exp(-\Delta t)] \end{aligned}$$

Thus, for large t,

$$E[x(t) - x(t + \Delta t)]^2 = V[\Delta t - (1-\exp-\Delta t)] \tag{D.34}$$

If furthermore Δt is small the above may be approximated by

$$E[x(t) - x(t + \Delta t)]^2 = \tfrac{1}{2}V\,(\Delta t)^2 \tag{D.35}$$

Change of variable formula for the Dirac delta function.

If in equation (D.5) we change the variable of integration from t to

$$s = ft \tag{D.36}$$

where f is a positive constant, we obtain

$$\int_{-\infty}^{\infty} \delta(s/f)\,ds = f \tag{D.37}$$

Consequently,

$$\delta(s/f) = f\delta(s) \tag{D.38}$$

APPENDIX E

Selection of 4 GPS Satellites for Optional Position Determination

In section 6 of the main text it was shown that the measurement d_j to the j-th satellite is given by

$$d_j = \sqrt{(\bar{r}_j-\bar{r})^T(\bar{r}_j-\bar{r})} + b, \tag{E.1}$$

where $\bar{r}_j$ is the position vector of the j-th GPS satellite, $\bar{r}$ is the position vector of the user, and b is a measurement bias.

In this appendix we shall determine where the 4 GPS satellites ideally should be located in order that the user's position may be calculated with the least amount of error. To do that we make the following assumptions

$$E\ \delta\bar{r}_j = E\ \delta d_j = 0, \tag{E.2}$$

where $\delta\bar{r}_j$ is the error in the position of the j-th GPS satellite and δd_j is the measurement error. We further assume that the errors are uncorrelated:

$$E\ \delta\bar{r}_j\delta\bar{r}_k^T = 0,\ E\ \delta d_j\delta d_k = 0 \quad \text{for } j \neq k,$$

and

$$E\ \delta\bar{r}_j\delta\bar{r}_j^T = \bar{\sigma}^2\ I,\ E(\delta d_j)^2 = \sigma_0^2 \tag{E.3}$$

where $\bar{\sigma}$ and σ_0 are scalars, and I is the 3 x 3 identity matrix.

If the error in the calculated user position is denoted by $\delta\bar{r}$ and the error in the calculated bias by δb, then it follows from equation (E.1) that if the errors are small then

$$\delta d_j = v_j^T \; (\delta\bar{r}_j - \delta\bar{r}) + \delta b, \qquad \text{(E.4)}$$

where the unit vector v_j is defined by

$$v_j = (\bar{r}_j - \bar{r})/|\bar{r}_j - \bar{r}| \qquad \text{(E.5)}$$

Rearranging the terms in equation (E.4) we find that

$$v_j^T \; \delta\bar{r} - \delta b = s_j, \qquad \text{(E.6)}$$

where

$$s_j = v_j^T \delta\bar{r}_j - \delta d_j \qquad \text{(E.7)}$$

From equations (E.2) and (E.3) we deduce that

$$E\, s_j = 0, \qquad \text{(E.8)}$$

$$E\, s_j s_k = 0 \quad \text{if } j \neq k \qquad \text{(E.9)}$$

and

$$E\, s_j{}^2 = v_j^T \; \bar{\sigma}^2 \; v_j + \sigma_0{}^2$$

Since v_j is a unit vector the last equation reduces to

$$E\, s_j{}^2 = \sigma^2, \qquad \text{(E.10)}$$

where

$$\sigma^2 = \bar{\sigma}^2 + \sigma_0{}^2 \tag{E.11}$$

Combining the measurement from four GPS satellites, equation (E.6) may be rewritten as

$$A \begin{bmatrix} \delta\bar{r} \\ \delta b \end{bmatrix} = s, \tag{E.12}$$

where

$$A = \begin{bmatrix} v_1^T & , -1 \\ v_2^T & , -1 \\ v_3^T & , -1 \\ v_4^T & , -1 \end{bmatrix} \quad \text{and} \quad s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \tag{E.13}$$

It is interesting to note from equations (E.12) and (E.13) that the position error $\delta\bar{r}$ is a function of the 'user to GPS satellite' direction, but independent of the corresponding distance.

From equations (E.8), (E.9), (E.10), and (E.13) we deduce that

$$E\,s = 0 \quad \text{and}\; E\,ss^T = \sigma^2\, I, \tag{E.14}$$

where I is the 4 x 4 identity matrix.

Solving equation (E.12) we obtain

$$\begin{bmatrix} \delta\bar{r} \\ \delta b \end{bmatrix} = A^{-1}s \tag{E.15}$$

We hence deduce with the aid of equation (E.14) that

$$E \begin{bmatrix} \delta\bar{r} \\ \delta b \end{bmatrix} = 0 \tag{E.16}$$

and

$$E \begin{bmatrix} \delta\bar{r}\ \delta\bar{r}^T & \delta\bar{r}\delta b \\ \delta b\ \delta\bar{r}^T & \delta b^2 \end{bmatrix} = \sigma^2\ A^{-1}A^{-T} \tag{E.17}$$

It will now be shown that in order that the expected square of the position error (trace E $\delta\bar{r}\delta\bar{r}^T$) be a minimum, the angles between the four vectors v_1, v_2, v_3 and v_4 should all be equal.

Let

$$A^{-1} = \begin{bmatrix} B \\ h^T \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ h_1 & h_2 & h_3 & h_4 \end{bmatrix} , \tag{E.18}$$

where B is a 3 x 4 matrix; h is a 4-vector; b_1, b_2, b_3 and b_4 are 3-vectors; h_1, h_2, h_3 and h_4 are scalars.

Since $AA^{-1} = A^{-1}A = I$ it follows from equation (E.13) and (E.18) that

$$v_i^T\ b_j = \delta_{ij} + h_j \tag{E.19}$$

where δ_{ij} is the Kronecker delta,

$$\sum_{i=1}^{4} b_i\ v_i^T = I, \tag{E.20}$$

$$\sum_{i=1}^{4} b_i = 0, \tag{E.21}$$

$$\sum_{i=1}^{4} h_i = -1 \tag{E.22}$$

From Equations (E.17) and (E.18) we deduce that

$$\text{trace } E\ \delta\bar{r}\delta\bar{r}^T = \sigma^2 \text{ trace } BB^T \tag{E.23}$$

In order that the above quantity be a minimum it is necessary that

$$\frac{\partial}{\partial v_j}\left[\text{trace } BB^T + \sum_{i=1}^{4}(v_i^T v_i - 1)\lambda_i\right] = 0\ , \tag{E.24}$$

the Lagrangian multipliers λ_i having been introduced to take into account the fact that each v_i is a unit vector. The following result is derived at the end of this appendix

$$\frac{\partial}{\partial v_j} \text{trace } BB^T = -2\ BB^T\ b_j \tag{E.25}$$

Using equation (E.25) we deduce from equation (E.24) that

$$\lambda_j v_j = BB^T\ b_j \tag{E.26}$$

It follows from equations (E.21) and (E.26) that

$$\sum_{j=1}^{4} \lambda_j v_j = 0 \tag{E.27}$$

Premultiplying equation (E.26) by b_i^T we deduce with the aid of equation (E.19) that

$$b_i^T \; BB^T \; b_j = \lambda_j \; b_i^T \; v_j$$

$$= \lambda_j(\delta_{ij} + h_i) \qquad \text{(E.28)}$$

Since the left hand side of the above equation is symmetric in i and j we conclude that

$$\lambda_j \; h_i = \lambda_i \; h_j \qquad \text{(E.29)}$$

Summing the above equation with respect to i we deduce with the aid of equation (E.22) that

$$\lambda_j = -h_j \sum_{i=1}^{4} \lambda_i \qquad \text{(E.30)}$$

There does not appear to be any reason why λ_j should differ from λ_k. We therefore make the assumption (later to be justified, of course) that for all j,

$$\lambda_j = \lambda \qquad \text{(E.31)}$$

It hence follows from equation (E.30) that

$$h_j = -\tfrac{1}{4} \qquad \text{(E.32)}$$

Since by equation (E.18)

$$BB = \sum_{j=1}^{4} \; b_j b_j^T \qquad \text{(E.33)}$$

we conclude from equations (.31), (E.20) and (E.26) that

$$\lambda I = BB^T \; BB^T \; . \tag{E.34}$$

Since BB^T is a semi-positive definitive matrix this is only possible if

$$BB^T = \sqrt{\lambda} \; I \tag{E.35}$$

It hence follows from equation (E.26) that

$$b_j = \sqrt{\lambda} \; v_j \; . \tag{E.36}$$

From the above and equation (E.32) and (E.19) we conclude that

$$\sqrt{\lambda} \; {v_i}^T \; v_j = \delta_{ij} - \tfrac{1}{4} \tag{E.37}$$

Since v_i is a unit vector it follows that ($i = j$ in the above equation)

$$\sqrt{\lambda} = \tfrac{3}{4} \; . \tag{E.38}$$

Hence if $i \neq j$,

$${v_i}^T \; v_j = - \tfrac{1}{3} \tag{E.39}$$

This is the desired result. We conclude from equations (E.17), (E.18), (E.35), (E.38), (E.32) and (E.21) that

$$E \begin{bmatrix} \delta\bar{r}\delta\bar{r}^T & \delta\bar{r}\delta b \\ \delta b\delta\bar{r}^T & \delta b^2 \end{bmatrix} = \sigma^2 \begin{bmatrix} \tfrac{3}{4}I & o \\ o & \tfrac{1}{4} \end{bmatrix} \tag{E.40}$$

A set of unit vectors satisfying equation (E.39) are given by

$$
\begin{aligned}
v_1^T &= (\; 0 \;, \; 0 \;, \; 1 \;) \\
v_2^T &= (\; 2\sqrt{2}/3 \;, \; 0 \;, \; -1/3 \;) \\
v_3^T &= (\; -\sqrt{2}/3 \;, \; \sqrt{2/3} \;, \; -1/3 \;) \\
v_4^T &= (\; -\sqrt{2}/3 \;, \; -\sqrt{2/3} \;, \; -1/3 \;)
\end{aligned}
\qquad \text{(E.41)}
$$

By equations (E.36) and (E.38)

$$b_j = \tfrac{3}{4} \, v_j \qquad \text{(E.42)}$$

With h_j given by equation (E.32), equations (E.19) through (E.22) are readily verified.

From equations (E.33), (E.20) and (E.42) it follows that

$$BB^T = \tfrac{3}{4} \, I, \qquad \text{(E.43)}$$

which is consistent with equations (E.26), (E.42), (E.31) and (E.38). The solution set (E.41) is thus justified. Note that the angles between the vectors v_j equal $\cos^{-1}(-1/3)$ or 109.5 degrees, and that the expected square of the position error as given by equations (E.23) and (E.43)

$$E \, \delta\bar{r}^T \delta\bar{r} = \tfrac{9}{4} \, \sigma^2 \qquad \text{(E.44)}$$

E.1 The Selection of Two Satellites when two have already been Selected.

Let us assume that we are given two vectors v_1 and v_2. There is no loss of generality in assuming that they are of the form

$$\begin{aligned} v_1^T &= (\cos p, \sin p, o) \\ v_2^T &= (\cos p, -\sin p, o) \end{aligned} \tag{E.45}$$

For reasons of symmetry it follows that v_3 and v_4 must be of the form

$$\begin{aligned} v_3^T &= (\cos q, 0, \sin q) \\ v_4^T &= (\cos q, 0, -\sin q), \end{aligned} \tag{E.46}$$

for some angle q to be determined. From the above and equation (E.13) it follows that

$$A = \begin{bmatrix} \cos p, & \sin p, & 0, & -1 \\ \cos p, & -\sin p, & 0, & -1 \\ \cos q, & 0, & \sin q, & -1 \\ \cos q, & 0 & -\sin q, & -1 \end{bmatrix} \tag{E.47}$$

Hence

$$A^TA = \begin{bmatrix} 2(\cos^2 p+\cos^2 q), & 0 & , & 0 & , & -2(\cos p+\cos q) \\ 0 & , 2\sin^2 p, & & 0 & , & 0 \\ 0 & , & 0 & , 2\sin^2 q, & & 0 \\ -2(\cos p+\cos q) & , & 0 & , & 0 & , & 4 \end{bmatrix} \tag{E.48}$$

Inverting the above matrix we obtain

$$A^TA^{-T} = \begin{bmatrix} \frac{1}{(\cos p-\cos q)^2} & 0 & 0 & \frac{\cos p+\cos q}{2(\cos p-\cos q)^2} \\ 0 & \frac{1}{2\sin^2 p} & 0 & 0 \\ 0 & 0 & \frac{1}{2\sin^2 q} & 0 \\ \frac{\cos p+\cos q}{2(\cos p-\cos q)^2} & 0 & 0 & \frac{\cos^2 p+\cos^2 q}{(\cos p-\cos q)^2} \end{bmatrix} \tag{E.49}$$

Comparing the above with equation (E.18) we note that

$$\text{trace } BB^T = \frac{1}{(\cos p-\cos q)^2} + \frac{1}{2\sin^2 p} + \frac{1}{2\sin^2 q} \tag{E.50}$$

Since BB^T is proportional to the expected square of the position error, we choose q such as to minimize that trace. Letting the partial derivative with respect to q vanish we deduce that

$$\frac{2\sin q}{(\cos p-\cos q)^3} + \frac{\cos q}{\sin^3 q} = 0 \tag{E.51}$$

Let $\cos p = c$ and $\cos q = x$ (E.52)

Corresponding to equation (E.51) we then obtain

$$f(x,c) = 0, \tag{E.53}$$

where

$$f(x,c) = 2(1-x^2)^2 + x(c-x)^3 \tag{E.54}$$

In finding the solutions of the above equations, we may assume that

$$c > 0, \tag{E.55}$$

since this only involves the definition of the coordinate axes. Obviously, $c < 1$. From equation (E.54) we find that

$$f(-\infty,c) = +\infty$$
$$f(-1,c) = -(c+1)^3$$
$$f(0,c) = 2$$
$$f(1,c) = -(1-c)^3$$
$$f(\infty,c) = +\infty$$

From the above it is evident that equation (E.53) always has 4 real solutions, and of those there is always one and only one in the interval (-1,0). Since cosp is positive it is quite obvious from equation (E.50) that the desired solution is negative. Although it is not simple to obtain an exact solution, a good approximation is given by

$$x = g(c), \tag{E.56}$$

where

$$g(c) = .327c - .765 \tag{E.57}$$

the linear approximation being based on the exact solutions for $c = 1 \left[x = (\sqrt{17}-5)/2 \right]$ and $c = 0 \left[x = -\sqrt{2-\sqrt{2}} \right]$.

Let Δ^2 denote the expected square of the position error. From equations (E.23), (E.50) and (E.52) we obtain,

$$\Delta^2/\sigma^2 = \frac{1}{(c-x)^2} + \frac{1}{2(1-c)^2} + \frac{1}{2(1-x^2)} \tag{E.58}$$

A comparison of exact and approximate values of x as functions of c are given in table E.1. Formula (E.58) has also been evaluated in the table. Note that the optimum value for $c = .577 = 1/\sqrt{3}$ corresponds to solution (E.41).

TABLE E.1 Exact and Approximate Values of x, and the Expected Square of the Position Error as a Function of c.

c	x(exact)	x = g(c)	θ_{12}	θ_{34}	Δ^2/σ^2
.000	-.765	-.765	180.0	80.2	3.41
.174	-.710	-.708	160.0	89.9	2.80
.342	-.655	-.653	140.0	98.5	2.45
.500	-.603	-.602	120.0	106.0	2.27
.643	-.555	-.555	100.0	112.6	2.27
.766	-.515	-.515	80.0	118.0	2.50
.866	-.482	-.482	60.0	122.4	3.20
.940	-.458	-.458	40.0	125.5	5.44
.985	-.443	-.443	20.0	127.4	17.91
1.000	-.438	-.438	0.0	128.0	∞
.577	-.577	-.576	109.5	109.5	2.25

θ_{12} is the angle between satellites 1 and 2 as seen from the user.

θ_{34} is the angle between satellites 3 and 4 as seen from the user.

Δ^2 is the expected error of the square of the position error

σ^2 is the sum of the measurement error variance and the variance of the GPS position error measured along any axis.

E.2 Selection of the Fourth Satellite When Three Have Already Been Selected.

In equation (E.13) let

$$A_0 = \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix} \quad \text{and} \quad d_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{E.59}$$

Then

$$A = \begin{bmatrix} A_0 , & -d_0 \\ v_4^T , & -1 \end{bmatrix} , \tag{E.60}$$

Let

$$A^{-1} = \begin{bmatrix} B_0 , & b_4 \\ h_0^T & -h_4 \end{bmatrix} , \tag{E.61}$$

Then

$$A_0 B_0 - d_0 h_0^T = I \tag{E.62}$$

$$A_0 b_4 + d_0 h_4 = 0 \tag{E.63}$$

$$v_4^T B_0 - h_0^T = 0 \tag{E.64}$$

$$v_4^T b_4 + h_4 = 1 \tag{E.65}$$

From equation (E.63) we obtain

$$b_4 = A_0^{-1} d_0 h_4 . \tag{E.66}$$

From the above and equation (E.65) we find that

$$h_4 = (1 - v_4^T A_0^{-1} d_0)^{-1} \tag{E.67}$$

From equation (E.62) and (E.64) we obtain

$$v_4^T (A_0^{-1} + A_0^{-1} d_0 h_0^T) - h_0^T = 0.$$

Hence,

$$h_0^T = (1-v_4^T A_0^{-1} d_0)^{-1} v_4^T A_0^{-1}$$

$$= h_4 v_4^T A_0^{-1} \tag{E.68}$$

From the above and equation (E.62) we deduce that

$$B_0 = A_0^{-1} + A_0^{-1} d_0 v_4^T A_0^{-1} h_4 \tag{E.69}$$

Writing

$$e_0 = A_0^{-1} d_0 \quad \text{and} \quad f = A_0^{-T} v_4 , \tag{E.70}$$

equations (E.66), (E.67) and (E.69) may be rewritten in the form

$$b_4 = - e_0 h_4 \tag{E.71}$$

$$h_4 = (1-v_4^T e_0)^{-1} \tag{E.72}$$

and

$$B_0 = A_0^{-1} + e_0 f^T h_4 \tag{E.73}$$

Comparing equation (E.18) and (E.61) we see that

$$B = [B_0 , b_4] \tag{E.74}$$

It hence follows from equations (E.71) and (E.73) that

$$BB^T = A_0^{-1}A_0^{-T} + A_0^{-1}f\ e_0^T h_4 + e_0 f\ A_0^{-T}\ h_4$$

$$+ e_0 f^T f\ e_0^T\ h_4^2 + e_0 e_0^T\ h_4^2 \qquad \text{(E.75)}$$

Hence

$$\text{trace } BB^T = \text{trace } A_0^{-1}A_0^{-T} + 2h_4 e_0^T A_0^{-1} f$$

$$+ h_4^2 e_0^T\ e_0 (1 + f^T f) \qquad \text{(E.76)}$$

Since the expected square of the position error is proportional to trace BB^T, it follows that we must choose the fourth satellite such that

$$2h_4 e_0^T\ A_0^{-1} f + h_4^2\ e_0^T e_0 (1 + f^T\ f)$$

is a minimum. A_0 as defined by equation (E.59) may be inverted using the formula

$$A_0^{-1} = (v_2 \wedge v_3,\ v_3 \wedge v_1,\ v_1 \wedge v_2)/[(v_1 \wedge v_2) \cdot v_3] \qquad \text{(E.77)}$$

E.3 Derivation of Equation (E.25)

The desired formula is most easily derived using the customary index summation convention. In what follows Latin indices will assumethe values 1,2,3,4 and Greek indices the values 1,2,3. Since B is a 3 x 4 matrix it follows that

$$(BB^T)_{\alpha\beta} = B_{\alpha i}B_{\beta i} \qquad \text{(E.78)}$$

Hence

$$\text{trace } (BB^T) = B_{\alpha i}B_{\alpha i} \qquad \text{(E.79)}$$

From equation (E.18) we deduce that

$$(BA)_{\alpha\beta} = \delta_{\alpha\beta} \tag{E.80}$$

Consequently

$$B_{\alpha k}A_{ks} = \delta_{\alpha s}, \tag{E.81}$$

and

$$\frac{\partial}{\partial A_{j\mu}} (B_{\alpha k}A_{ks}) = 0 \tag{E.82}$$

Hence

$$\left(\frac{\partial}{\partial A_{j\mu}} B_{\alpha k}\right) A_{ks} + B_{\alpha j}\delta_{s\mu} = 0 \tag{E.83}$$

Post multiplying the above equation by A^{-1}_{si} we deduce that

$$\frac{\partial}{\partial A_{j\mu}} B_{\alpha i} + B_{\alpha j}A^{-1}_{\mu i} = 0 \tag{E.84}$$

i.e.

$$\frac{\partial}{\partial A_{j\mu}} B_{\alpha i} = -B_{\alpha j}B_{\mu i} \tag{E.85}$$

$$B_{\alpha i} \frac{\partial}{\partial A_{j\mu}} B_{\alpha i} = - B_{\alpha j}B_{\mu i}B_{\alpha i}, \tag{E.86}$$

and

$$\frac{\partial}{\partial A_{j\mu}} \text{trace } BB^T = -2\, B_{\alpha j}B_{\mu i}B_{\alpha i} \tag{E.87}$$

In accordance with equations (E.13) and (E.18) this may also be written as

$$\left(\frac{\partial}{\partial v_j} \text{trace } BB^T\right)_\mu = -\, 2\, (BB^T)_{\mu\alpha}(b_j)_\alpha \tag{E.88}$$

Hence

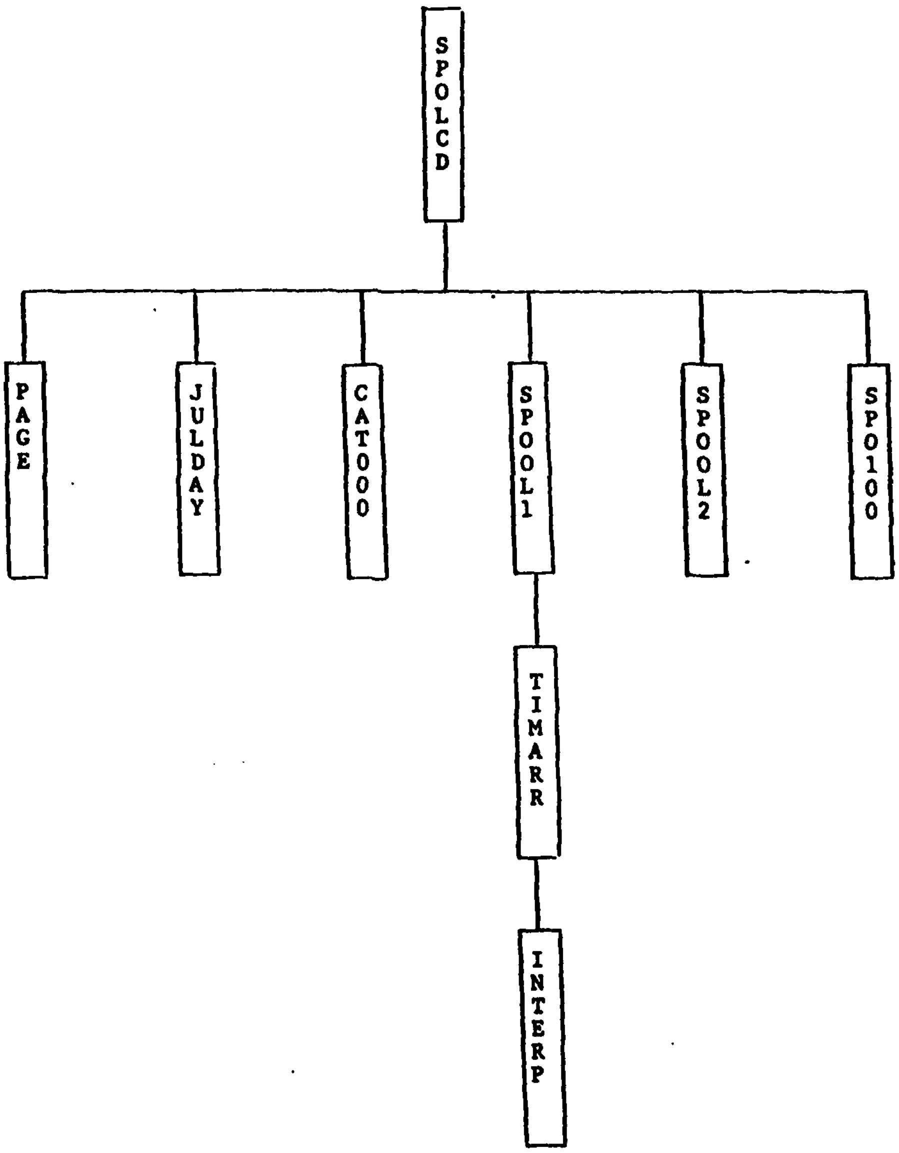
$$\frac{\partial}{\partial v_j} \text{ trace } BB^T = -2BB^T b_j, \tag{E.25}$$

the desired result.

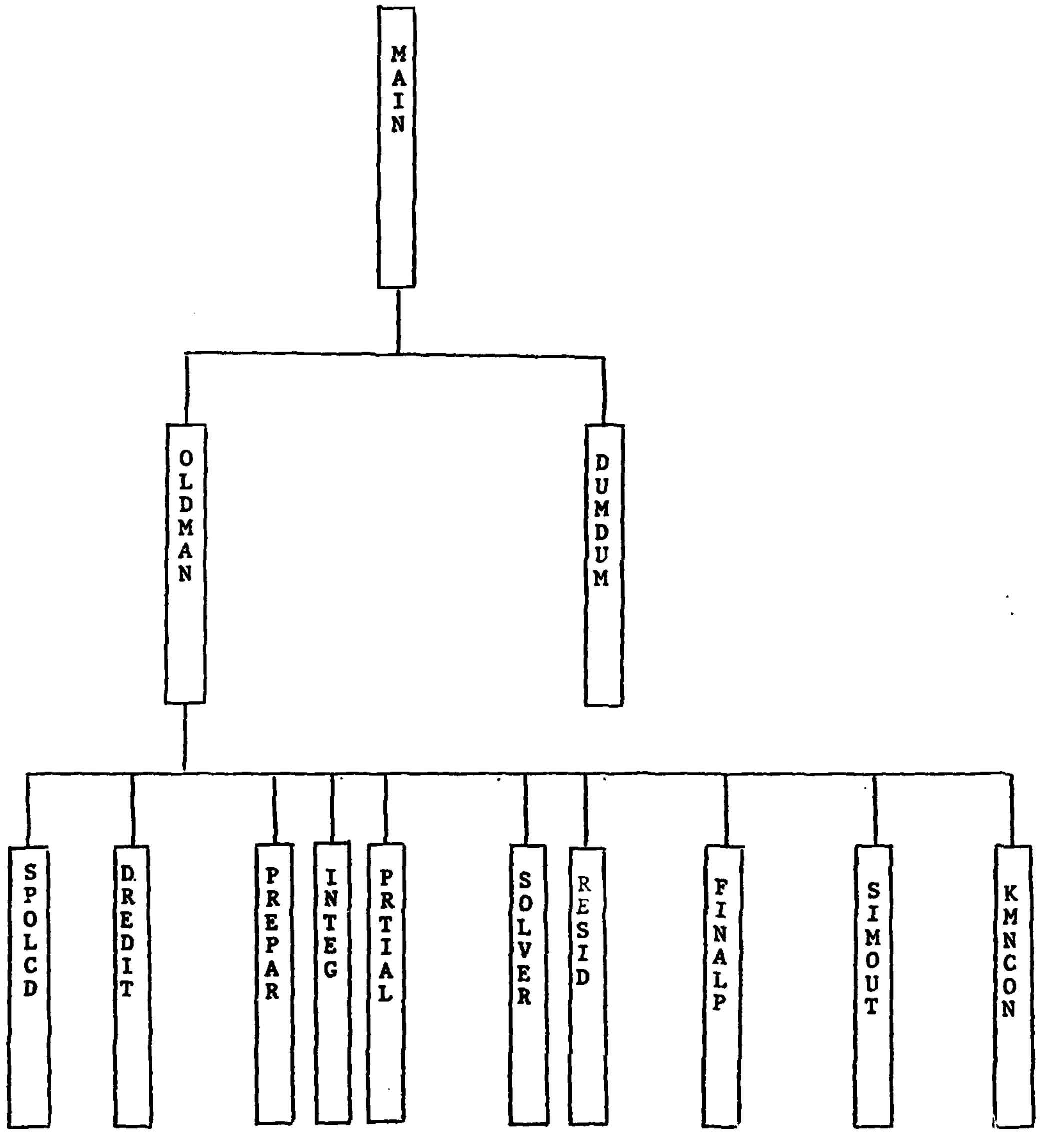
APPENDIX F

Flow Charts and Short Descriptions of Fotonap Subroutines

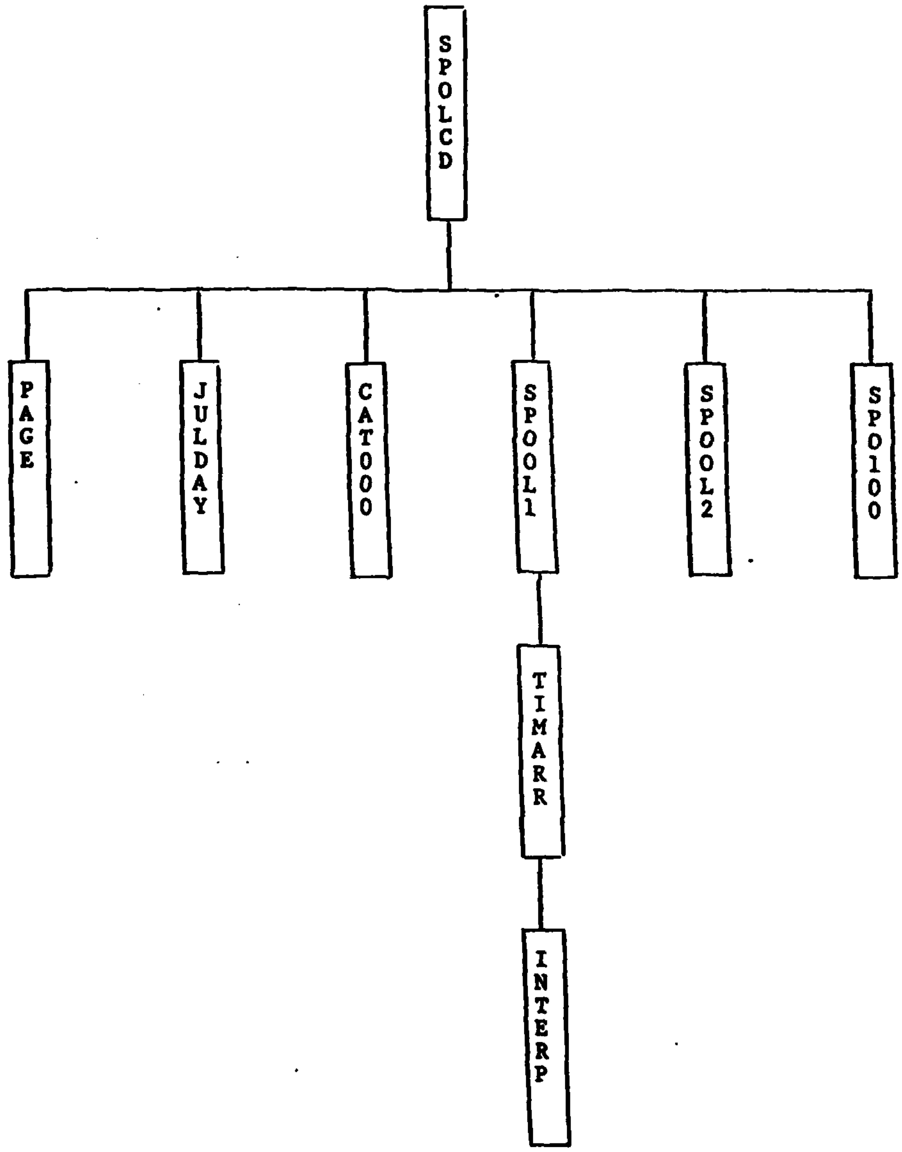
Flow Chart F.2



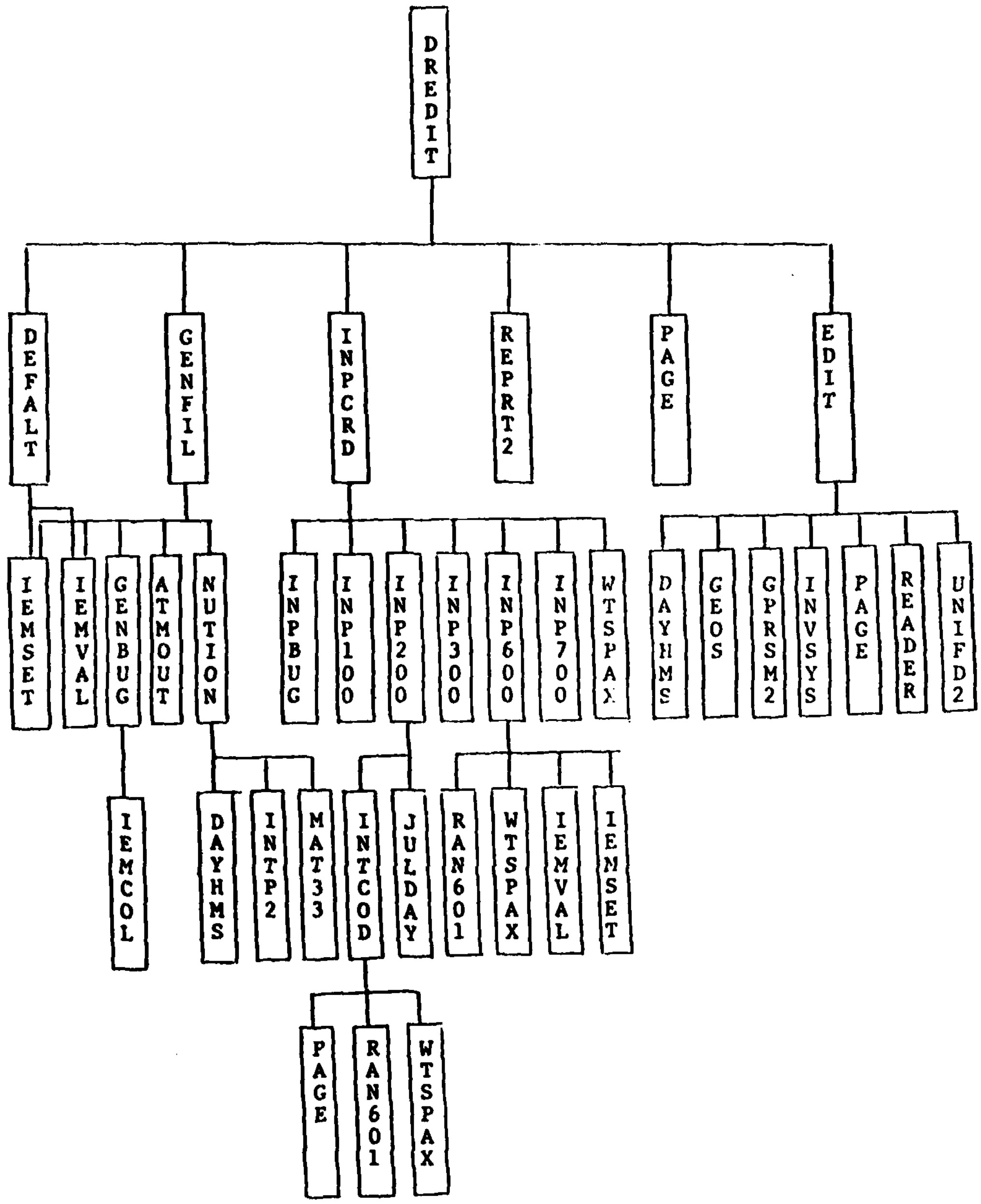
Flow Chart F.1



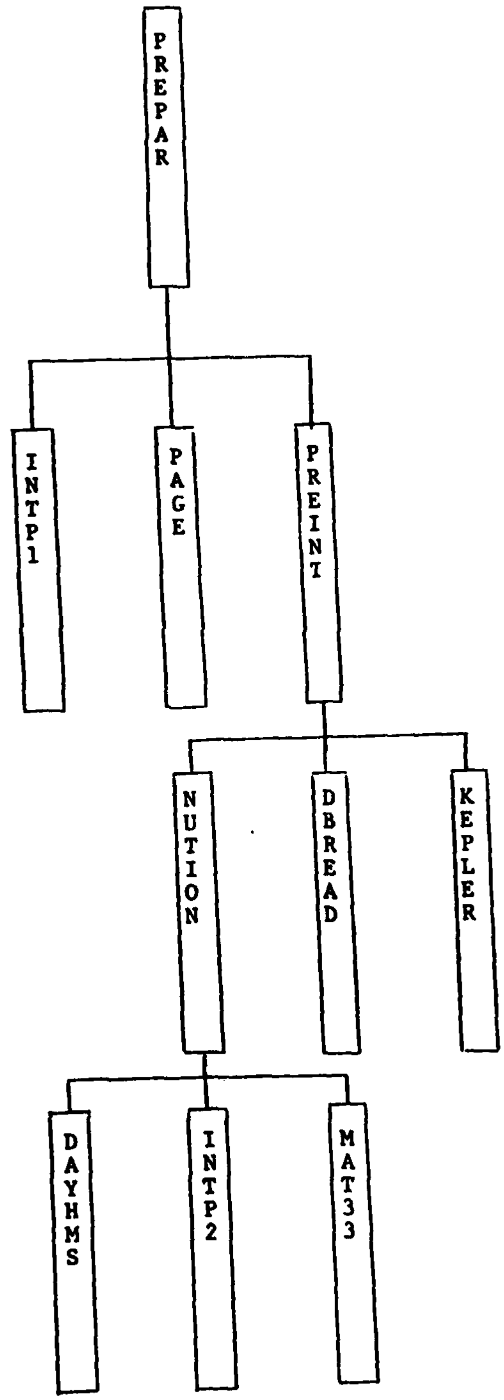
Flow Chart F.2



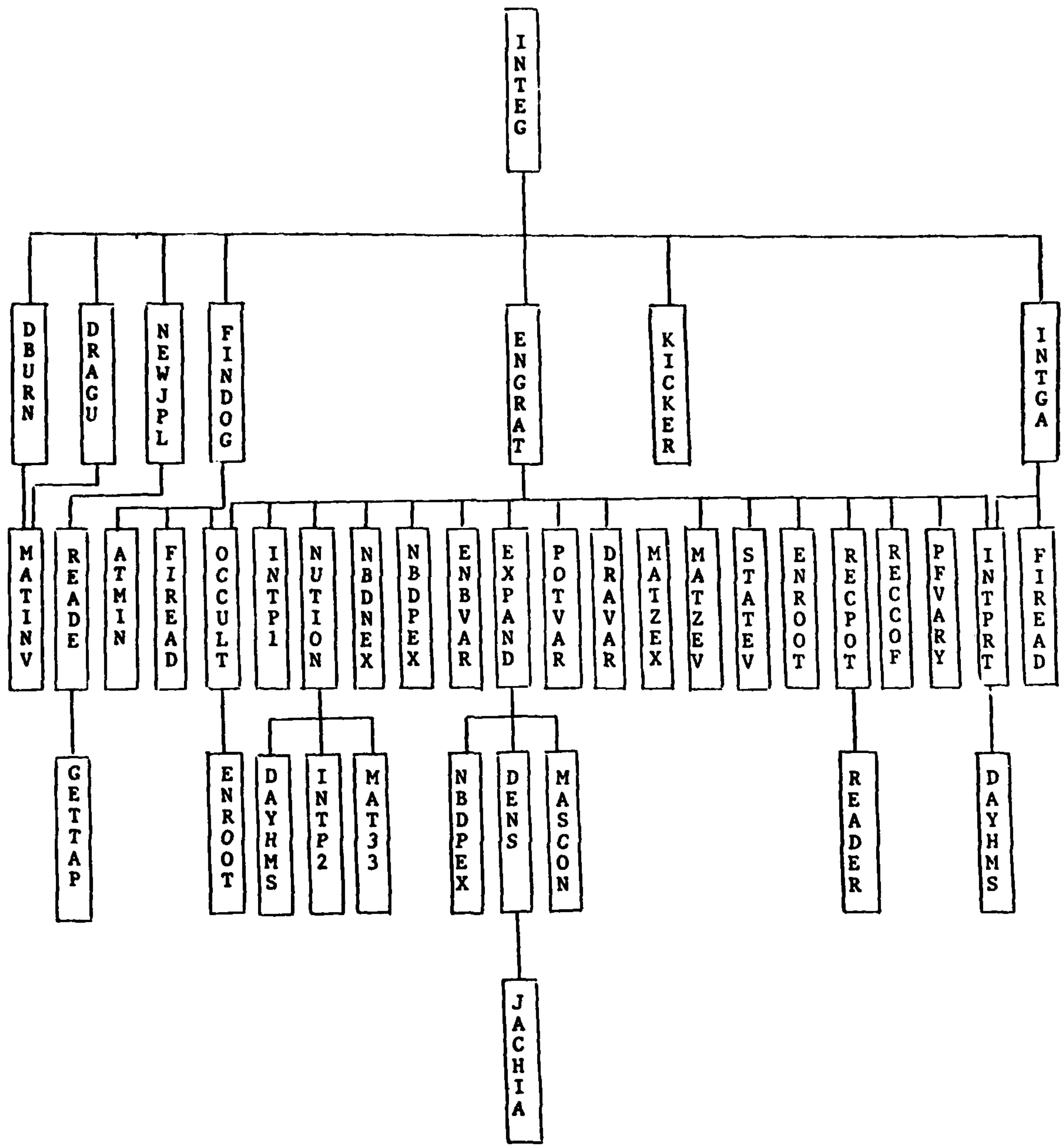
Flow Chart F.3



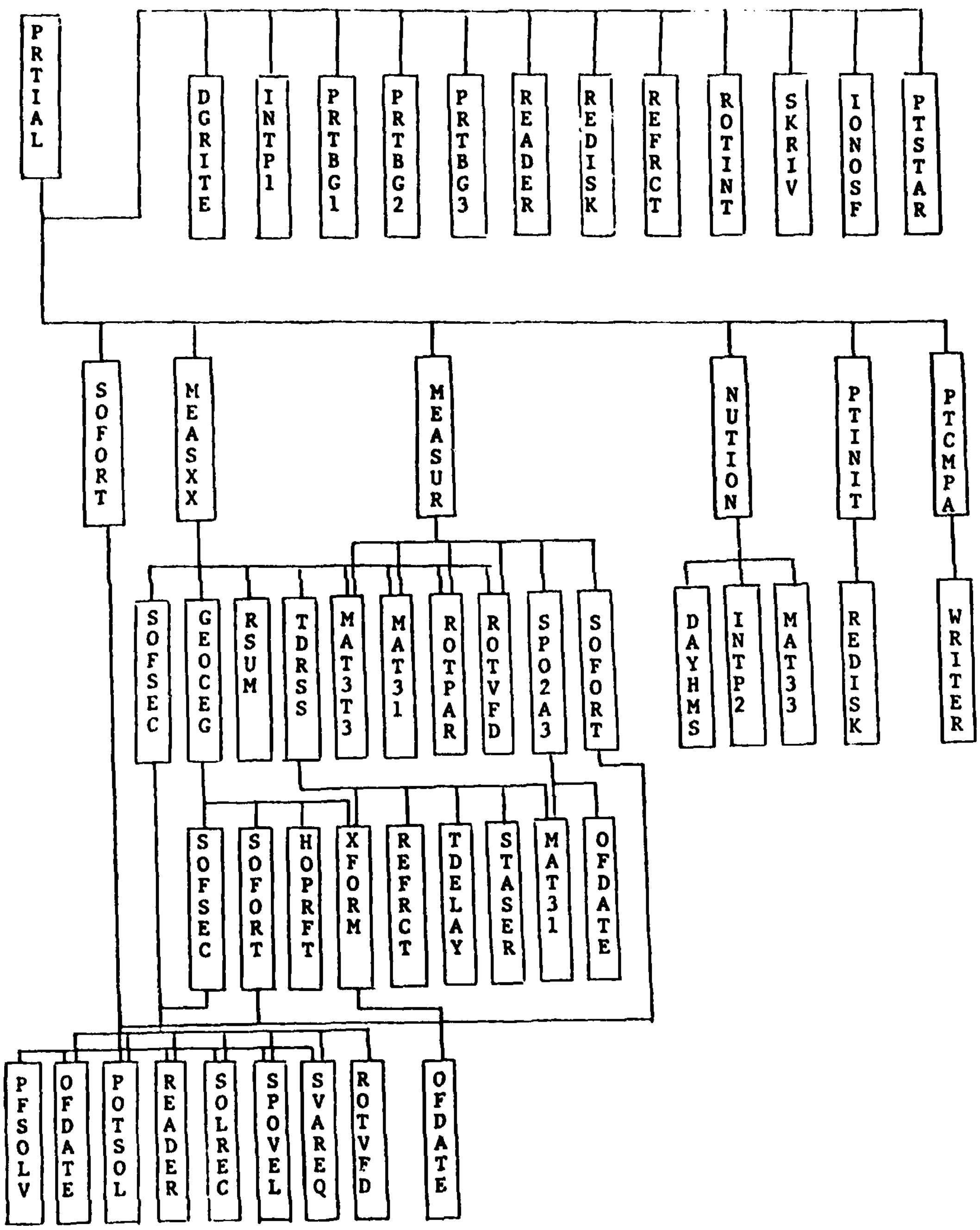
Flow Chart F.4



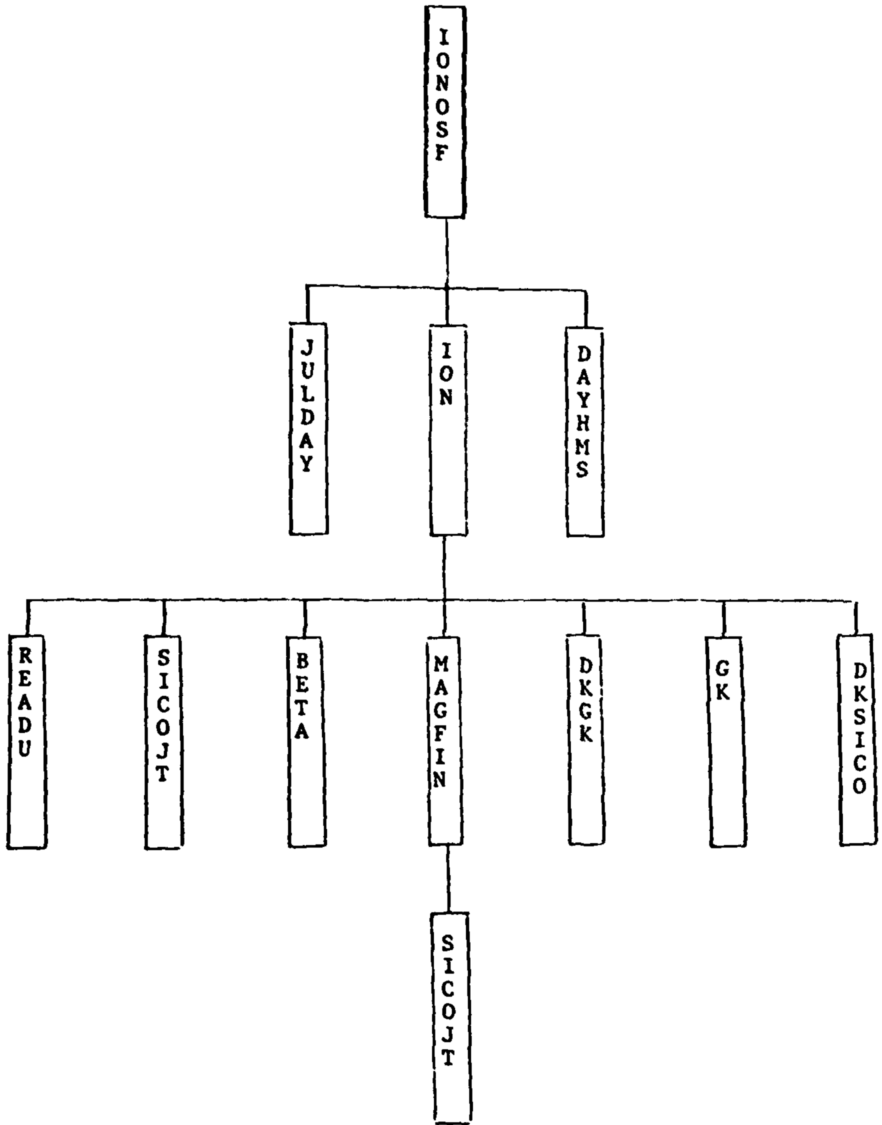
Flow Chart F.5



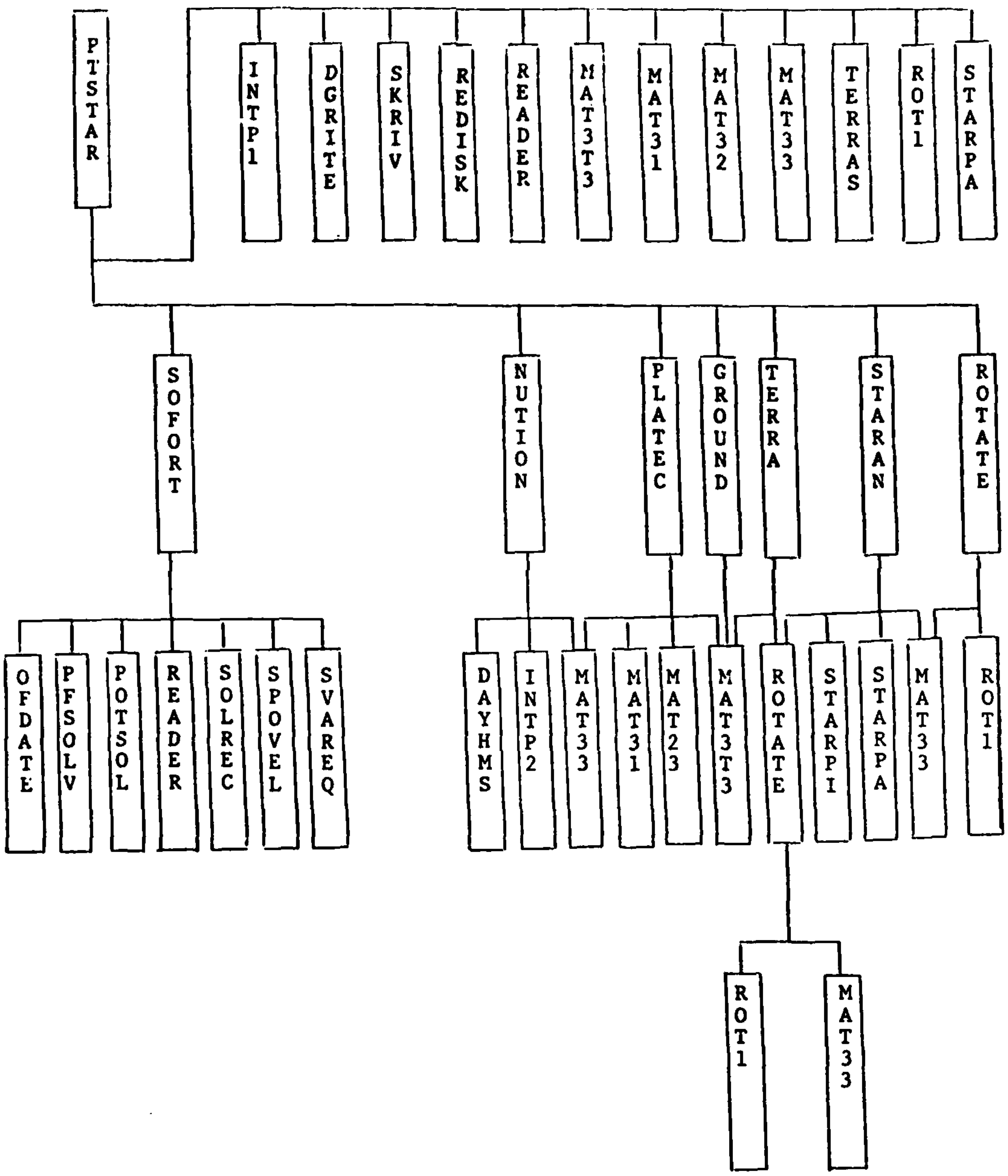
Flow Chart F.6



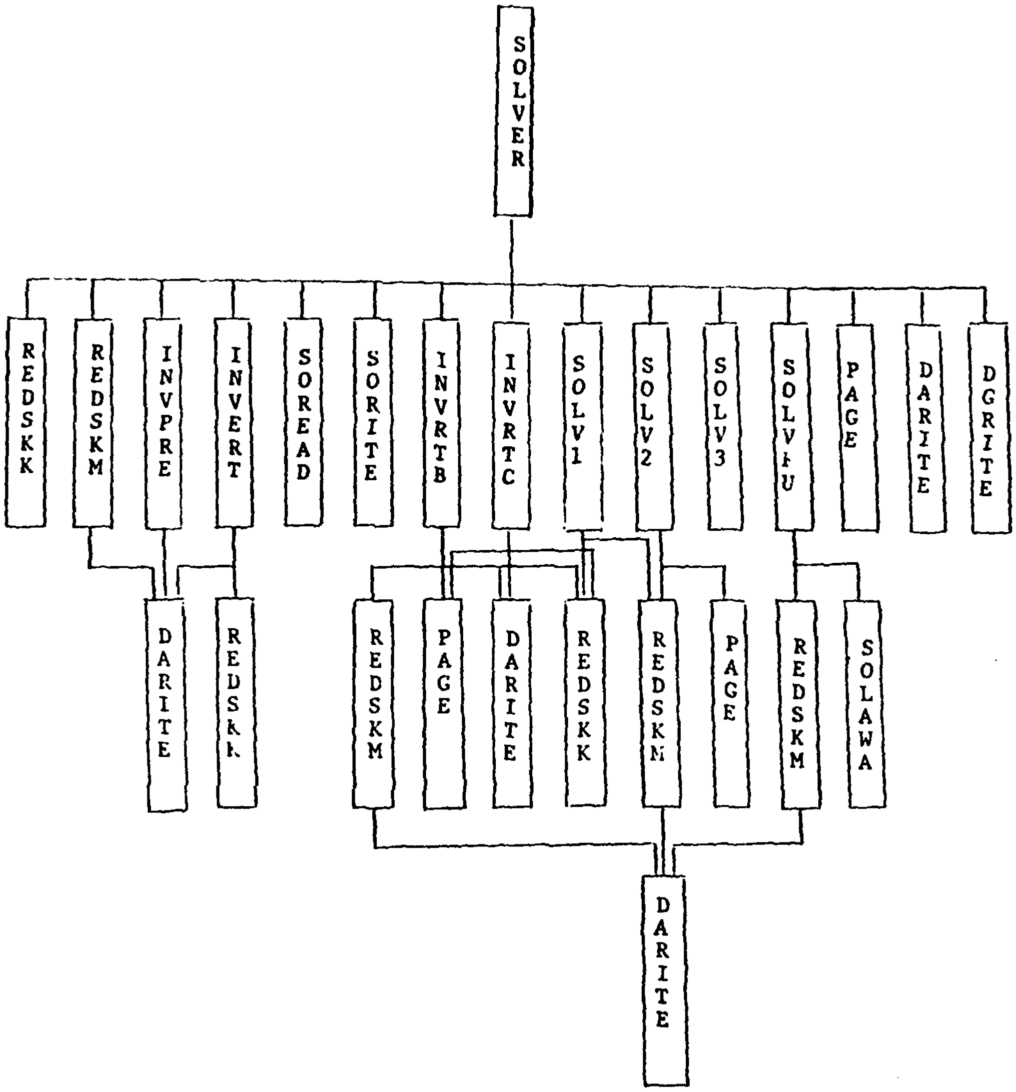
Flow Chart F.7



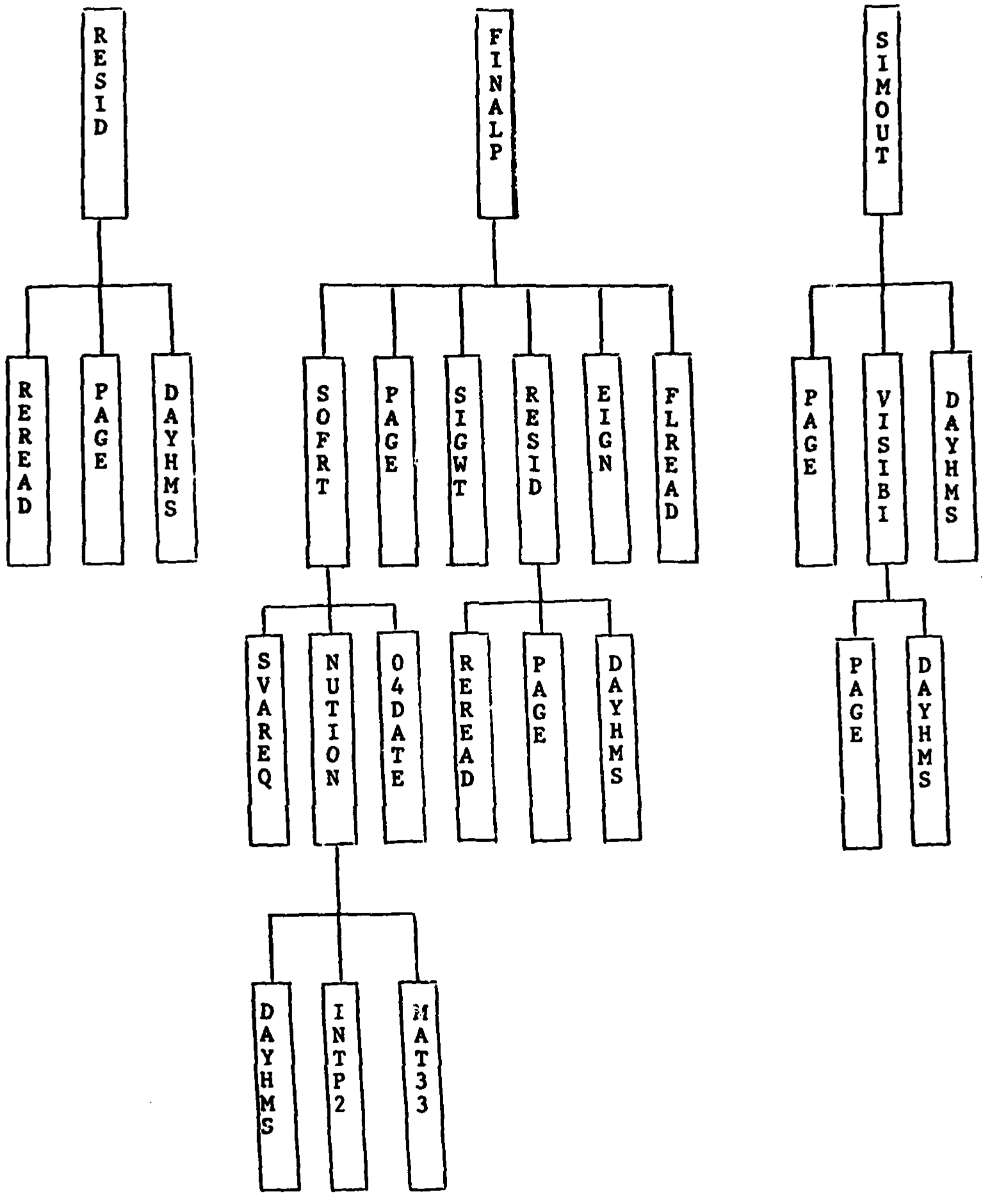
Flow Chart F.8



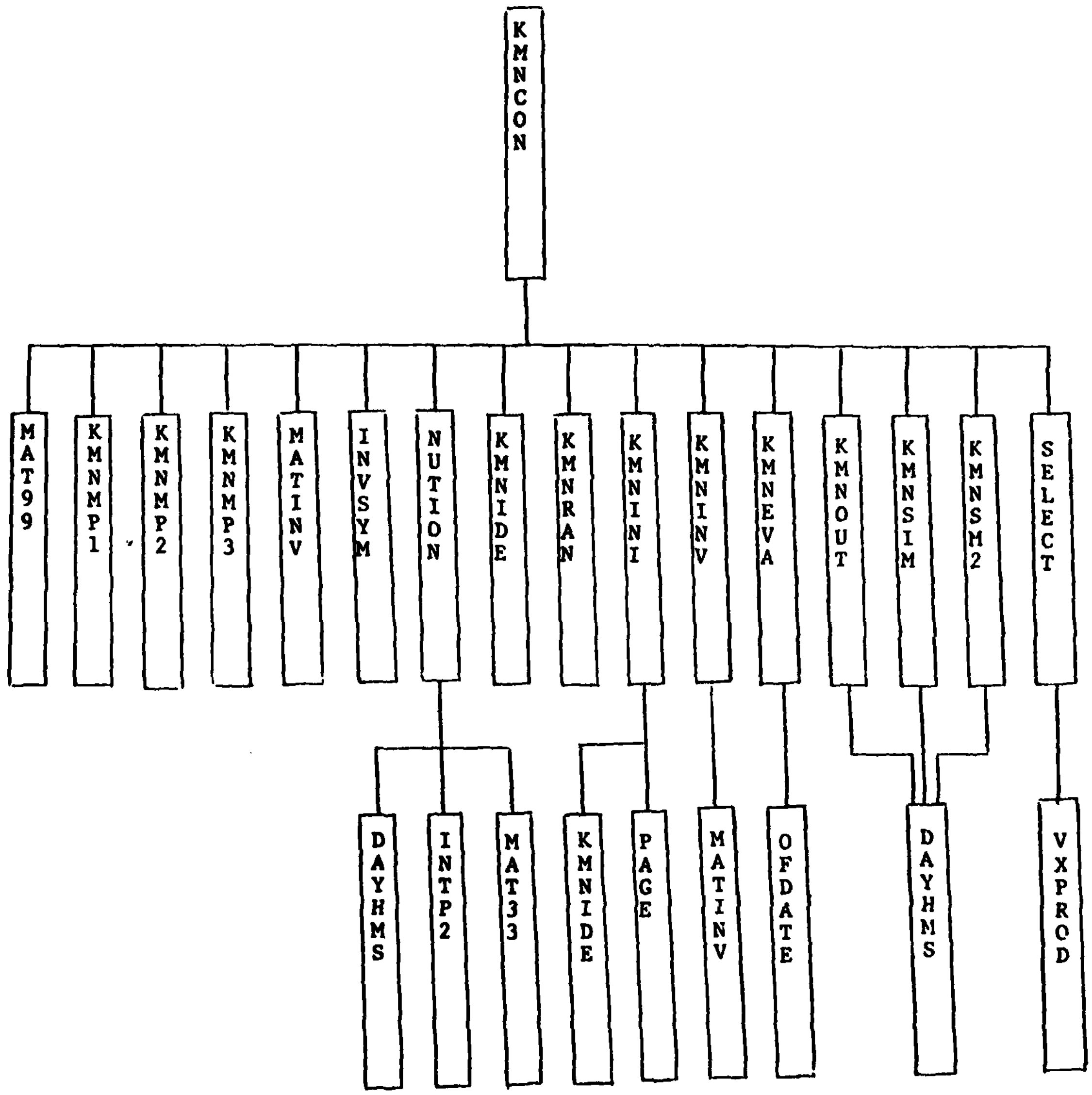
Flow Chart F.9



Flow Chart F.10



Flow Chart F.11



ALPHABETIC LISTING AND SHORT DESCRIPTION OF PHOTONAP SUBROUTINES

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
ATMIN	FINDOG	Reads tabulated atmospheric densities from file LUA
ATMOUT	GENFIL	Writes tabulated atmospheric densities on file LUA
BETA	ION	Used in computation of ionospheric corrections
CAT000	SPOLCD	Converts "free form" card input to standard NAP format (not on 1108)
DARITE	SOLVER INVPRE REDSKM INVERT INVRTB INVRTC	Utility routine for direct access file 43
DAYHMS	NUTION EDIT INTPRT IONOSF RESID SIMOUT VISIBI KMNOUT KMNSIM KMNSM2	Converts (Julian day, seconds of day) to (year, month, day, hour, min, sec)
DBREAD	PREINT	Utility routine for direct access file 41
DBURN	INTEG	Adds velocity increment to satellite velocity (Discrete thrust)
DEFALT	DREDIT	Initializes program constants to their default values (See User's Guide)
DENS	EXPAND	Computes atmospheric density as function of height (*)
DGRITE	PTSTAR SOLVER PRTIAL	Utility routine for direct access file 40

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
DKGK	ION	Used in computation of ionospheric corrections
DKSICO	ION	Used in computation of ionospheric corrections
DRAGU	INTEG	Called at the end of each drag segment (except the last) to calculate the appropriate partial derivatives
DRAVAR	ENGRAT	Computes drag contribution to variational equations (*)
DREDIT	OLDMAN	Control routine for processing control card input
DUMDUM	PHOTO	Dummy routine used for switching program overlays
EDIT	DREDIT	Edits observed data or (in simulation mode) generates random numbers
EIGN	FINALP	Computes eigen-values of a matrix
ENBVAR	ENGRAT	Computes central term and planetary contribution to variational equations (*)
ENGRAT	INTEG	Control routine for each integration step (*)
ENROOT	ENGRAT OCCULT	Finds the zero of a function expressed as a power series
EXPAND	ENGRAT	Develops power series coefficients for satellite vector (*)
FINALP	OLDMAN	Prints final results
FINDOG	INTEG	Initializes integrator at start of integration or on change of origin
FIREAD	FINDOG INTGA	Utility routine for direct access file 41
FLREAD	FINALP	Utility routine for direct access file 41

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
GENBUG	GENFIL	Debug print for subroutine GENFIL (formerly part of GENFIL)
GENFIL	DREDIT	Generates internal files based on control card input
GEOCEG	MEASXX	Routine for handling geoceiver measurements
GEOS	EDIT	Reads data tape in GEOS format
GETTAP	READE	Reads planetary ephemerides from file (10)
GK	ION	Used in computation of ionospheric corrections
GPRSM2	EDIT	Reads data tape in NAP format
GROUND	PTSTAR	Estimates ground point coordinates by projecting photographic plate coordinates onto Earth's surface
HOPRFT	GEOCEG	Tropospheric refraction corrections for geoceiver data
IEMCOL	GENBUG	Subroutine for unpacking integers and storing them (unpacked) in an array
IEMSET	DEFALT INP600 GENFIL	Function for packing integers
IEMVAL	DEFALT INP600 GENFIL	Function of unpacking integers
INPBUG	INPCRD	Called by INPCRD for debug print
INPCRD	DREDIT	Processes control card input
INP100	INPCRD	Used for processing series 100 input cards (formerly part of routine INPCRD)

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
INP200	INPCRD	Used for processing series 200 input cards (formerly part of routine INPCRD)
INP300	INPCRD	Used for processing series 300 input cards (formerly part of subroutine INPCRD)
INP600	INPCRD	Used for processing series 600 input cards (formerly part of routine INPCRD)
INP700	INPCRD	Used for processing series 700 input cards (formerly part of routine INPCRD)
INTCOD	INP200	Generates arrays for recovery of gravity parameters (spherical harmonics or mascons)
INTEG	OLDMAN	Control routine for integrator
INTERP	TIMARR	Interpolation routine used for setting up tables of time corrections
INTGA	INTEG	Prints integrator output at end of integration of each arc
INTPRT	ENGRAT INTGA	Prints time corresponding to integrator output
INTP1	PREPAR ENGRAT PRTIAL PTSTAR	Estimates difference between integrator time and UTC through interpolation
INTP2	NUTION	Estimates difference between UT1 and integrator time through interpolation
INVERT	SOLVER	Part 1 of matrix inversion
INVPRE	SOLVER	Used in processing photogrammetric data. (i) resequences file 26 (IPASOT) of ground point images on file 38. (ii) on first iteration writes ground point records from file 25 (IARCOT) to file 23 (IGPT)
INVRTB	SOLVER	Part 1 of matrix inversion. Prints intermediate results

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
INVRTC	SOLVER	Similar to INVRTB, but solution obtained without completing matrix inversion
INVSYM	KMNCON	IN SITU inversion routine for positive definite matrix in upper diagonal form
INVSYS	EDIT	Similar to INVSYM. Additional feature is to check for singularity
ION	IONOSF	Used in computation of ionospheric corrections
IONOSF	PRTIAL	Used in computation of ionospheric corrections
JACHIA	DENS	Computes atmospheric density (Lockheed-Jacchia model)
JULDAY	SPOLCD INP200 IONOSF	Converts (year, month, day, hours, minutes, seconds) to (Julian day, seconds of day)
KEPLER	PREINT	Converts Keplerian to Cartesian input
KICKER	INTEG	Initializes integrator common blocks
KMNCON	OLDMAN	Control routine for Kalman filtering and smoothing
KMNEVA	KMNCON	Used in Kalman filtering for evaluating integrated power series
KMNIDE	KMNINI KMNCON	Store (9 x 9) identity matrix in required location
KMNINI	KMNCON	Initialization routine for Kalman filtering and smoothing
KMNINV	KMNCON	Used in Kalman filtering for calculating the transition matrix relative to the previous time point (given the transition matrices relative to epoch)
KMNMP1	KMNCON	Utility routine for computing $C = A * B^T$ where A is symmetric and stored in upper triangular form

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
KMNMP2	KMNCON	Utility routine for computing C = A * B, where C is symmetric. C is stored in upper triangular form
KMNMP3	KMNCON	Utility routine for computing $C = A^T * B * A$, and $x = A^T y$. B and C are symmetric and stored in upper triangular form
KMNOUT	KMNCON	Used to output the parameter estimates (and their covariances) from the Kalman filter and smoother. (Output on file 29 (ITAPE))
KMNRAN	KMNCON	Random number generator associated with simulations in Kalman filtering
KMNSIM	KMNCON	Simulates GPS output used in Kalman filtering (Output on file 33 (LF33))
KMNSM2	KMNCON	Printout routine for Kalman filtering. In simulation mode prints full 9 parameter state-vector. In filter or smoother mode prints only 3 parameter state-vector (first 6 parameters printed in KMNOUT)
MAGFIN	ION	Used in computation of ionospheric corrections
MAIN (see photo)		Control routine for NAP program
MASCON	EXPAND	Computes mascon contributions to satellite acceleration (**)
MATINV	KMNINV KMNCON DBURN DRAGU	Matrix inversion
MATZEV	ENGRAT	Evaluates state transition matrix at end of integration step (*)
MATZEX	ENGRAT	Develops coefficients for power series expansion of state transition matrix (*)

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
MAT23	PLATEC	Matrix multiplication (2x3)x(3x3)
MAT3T3	MEASUR MEASXX PLATEC GROUND PTSTAR TERRA	Matrix multiplication (3x3)'x(3x3) (' indicates transpose)
MAT31	MEASUR SPO2A3 MEASXX TDRSS PTSTAR PLATEC	Matrix multiplication (3x3)x(3x1)
MAT32	PTSTAR	Matrix multiplication (3x3)x(3x2)
MAT33	NUTION PTSTAR PLATEC STARAN ROTATE	Matrix multiplication (3x3)x(3x3)
MAT99	KMNCON	C = A * B. A is a (9x9) matrix, C and B are (9xn), where n = 9 (entry MAT99) or n = 1 (entry MAT91)
MEASUR	PRTIAL	Routine for handling the following measurement types: RANGE, AZIMUTH, ELEVATION, RIGHT ASCENSION, DECLINATION, MINITRACK DIRECTION COSINES, X30 and Y30 ANGLES, DISTANCE TO ELLIPSOID, RANGE RATE, MINITRACK RATES, X85 and Y85 ANGLES, STATE VECTOR MEASUREMENTS
MEASXX	PRTIAL	Routine for handling the following measurement types: RANGE SUM, RANGE SUM RATE, GRARR, TDRSS, GEOCEIVER
NBDNEX	ENGRAT	Develops coefficients for power series expansion of Sun, Moon, and Planets (*)
NBDPEX	ENGRAT EXPAND	Computes central term and planetary contribution to satellite acceleration (**)

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
NEWJPL	INTEG	Used in conjunction with READE and GETTAP to obtain planetary ephemerides
NUTION	GENFIL PREINT ENGRAT PRTIAL PTSTAR SOFRT KMNCON	Calculates precession/nutation matrix and Greenwich Hour Angle
OCCULT	ENGRAT FINDOG	If satellite is orbit around body A, this routine determines if satellite is visible from body B
OFDATE	SOFORT SPO2A3 SOFSEC XFORM KMNEVA	Rotates vector or matrix from "inertial 1950.0" to "true of date" (Double Precision)
OLDMAN	PHOTO	("old main") secondary control routine for NAP program
O4DATE	SOFRT	Rotates vector or matrix from "inertial 1950.0" to "true of date" (single precision)
PAGE	SPOLCD DREDIT EDIT INTCOD PREPAR RESID FINALP SIMOUT VISIBI KMNINI SOLVER SOLV2 INVRTB INVRTC	Prints page heading
PFSOLV	SOFORT	Evaluates partials of satellite vector w.r.t. continuous thrust parameters using previously computed power series

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
PFVARY	ENGRAT	Develops power series coefficients for partials of satellite vector w.r.t. continuous thrust parameters (*)
PLATEC	PTSTAR	Computes predicted photographic plate coordinates and range (Photogrammetric measurement types 7-9)
POTSOL	SOFORT SOFSEC	Evaluates partials of satellite vector w.r.t. gravity parameters using previously computed power series
POTVAR	ENGRAT	Computes central body (excluding central term--see ENBVAR) contribution to variational equations (*)
PREINT	PREPAR	Sets up arrays for integrator based on current values of "solve for" parameters
PREPAR	OLDMAN	Sets up arrays for integrator based on control files
PRTBG1	PRTIAL	Output debug print from subroutine PRTIAL
PRTBG2	PRTIAL	Output debug print from subroutine PRTIAL
PRTBG3	PRTIAL	Output debug print from subroutine PRTIAL
PRTIAL	OLDMAN	Computes differences between observations and predicted observations. Also computes associated partials. Results output on file (ISFILE)
PTCMPA	PRTIAL	Data compression associated with subroutine PRTIAL (formerly part of PRTIAL)
PTINIT	PRTIAL	Used for initializing variables used in subroutine PRTIAL (formerly part of PRTIAL)
PTSTAR	PRTIAL	Routine for handling photogrammetric measurements (formerly part of PARTIAL)

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
RAN601	INP600 INTCOD	Modifies a parameter value by adding a random Gaussian number with a given standard deviation
READE	NEWJPL	Used in conjunction with GETTAP to obtain planetary ephemerides
READER	EDIT RECPOT PRTIAL SOFORT SOFSEC PTSTAR	Utility routine for read/write from/to sequential file
READU	ION	Used in computation of ionospheric corrections
RECCOF	ENGRAT	Develops power series coefficients for partials of satellite w.r.t. a single parameter (Used for solar pressure and drag) (*)
RECPOT	ENGRAT	Develops power series coefficients for partials of satellite vector w.r.t. gravity parameters (*)
REDISK	PRTIAL PTINIT PTSTAR	Utility routine for read/write of totally stable parameters on random access file
REDSKK	SOLVER SOLV1 INVERT INVRTB INVRTC	Utility routine for read/write of totally stable parameters on random access file
REDSKM	SOLVER SOLVFU SOLV1 SOLV2 INVRTB	Utility routine for read/write of normal equation coefficient matrix on random access file
REFRCT	PRTIAL TDRSS	Computes tropospheric refraction corrections
REPRT2	DREDIT	Generates printed report of run conditions as specified by the control cards (temporarily removed from NAP)

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
REREAD	RESID	Utility routine for direct access file 41
RESID	OLDMAN FINALP	Prints measurement residuals
ROTATE	PTSTAR STARAN TERRA	Computes a rotation matrix corresponding to sequential rotations about principal axes
ROTINT	PRTIAL	Converts (latitude, longitude, height) to Cartesian coordinates. Computes rotation matrix "Earth fixed Geocentric to Local"
ROTPAR	MEASUR MEASXX	Rotates measurement partials w.r.t. satellite state-vector from "Earth fixed" to "True of date"
ROTVFD	MEASUR MEASXX SOFSEC	Rotates a vector (v) from "true of date" (D) to "Earth fixed" (F)
ROT1	PTSTAR ROTATE	Computes a rotation matrix corresponding to a rotation about a principal axis
RSUM	MEASXX	Computes predicted range sum and range sum rate measurements (Measurement types 16-17)
SELECT	KMNCON	Associated with GPS measurements. Computes GPS satellite position, user distance to them and partial derivatives w.r.t. user position. In simulation mode, selects an optimal set of 4 GPS satellites.
SICOJT	ION MAGFIN	Used in computation of ionospheric corrections
SIGWT	FINALP	Converts weights to standard deviations and vice versa
SIMOUT	OLDMAN	Computes simulated measurements. Outputs results on file
SKRIV	PRTIAL PTSTAR	Utility routine for writing data on sequential file

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
SOFORT	PRTIAL MEASUR GEOCEG PTSTAR	Reads power series for primary satellite state vector and partials from sequential file and evaluates at required time point
SOFRT	FINALP	Reads power series for primary satellite state-vector and partials from sequential file and evaluates partials w.r.t. the initial state-vector. Computes state-vector covariance matrix
SOFSEC	MEASXX GEOCEG	Reads power series for secondary satellite state-vector and partials from sequential file and evaluates at required time point
SOLAWA	SOLVFU	Function for computing A^TWA, A^TWy, y^TWy. A is a (6xn) matrix, y is a 6-vector. W is a (6x6) symmetric matrix stored in upper triangular form. (called from subroutine SOLVER when processing data from Kalman filter output)
SOLREC	SOFORT SOFSEC	Evaluates power series to obtain partials of satellite state-vector w.r.t. a parameter (solar pressure and drag)
SOLVER	OLDMAN	Control routine for generating and solving normal equations
SOLVFU	SOLVER	Used for computing the contribution of 6 correlated measurements to the Normal Equations Matrix and Vector
SOLV1	SOLVER	Used for computing contribution of a priori parameter values to Normal Equations Matrix and Vector (Subroutine SOLV1 was formerly part of subroutine SOLVER)
SOLV2	SOLVER	Prints correlation vector for each arc and stores primary arc covariance matrix (entry SOLV2). Stores parameter numbers and stability types for primary and secondary arcs (entry SOLV2A) Initializes normal equations matrix and vector for multiple drag segments (entry SOLV2B). (Subroutine SOLV2 was formerly part of subroutine SOLVER)

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
SOLV3	SOLVER	Used for deciding when to terminate iterative solution process (entry SOLV3). Prints summary associated with inversion of Normal Equations Matrix (entry SOLV3A) (Subroutine SOLV3 was formerly part of subroutine SOLVER)
SOREAD	SOLVER	Utility routine for reading a sequential file
SORITE	SOLVER	Utility routine for writing a sequential file
SPOLCD	OLDMAN	Scans NAP control cards for consistency. Generates some arrays and files based on the control cards. The control cards are reformatted and output on file (ICARD) for final processing by INPCRD
SPOOL1	SPOLCD	Used for computing interpolation tables for E.T., UTC and UT1 differences (via call to subroutine TIMARR), and computing and sorting time intervals for which integrator output is required. Tables and time intervals are temporarily stored in file 32 (LUB). (formerly part of subroutine SPOLCD)
SPOOL2	SPOLCD	Used only for photogrammetric data. Generates ground point labels for output on file 25 (IARCOT), and ground point coordinates output on direct access file 30, (I30). (entry SPOOL2) clears array for ground point coordinates (entry SPOOL3). (formerly part of subroutine SPOLCD)
SP0100	SPOLCD	Used for preliminary processing of 100 series cards (entry SP0100). Used for processing meteorological data used in refraction formulae (entry SP0610) and outputs processed data as well as TDRSS data on file 26 (IPASOT) (entry SPOOL4). (formerly part of SPOLCD)
SP02A3	MEASUR	Evaluate integrated power series output for 2nd and 3rd time derivatives of position

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
SPOVEL	SOFORT SOFSEC	Evaluates power series to obtain satellite state-vector
STARAN	PTSTAR	Computes predicted stellar camera orientation angles (photogrammetric measurement types 1-6)
STARPA	PTSTAR STARAN	Computes a (3x3) matrix (See note 1)
STARPI	STARAN	Computes a (3x3) matrix (See note 1)
STASER	TDRSS	Develops power series coefficients for station vector in inertial space at a specified time T. The intertial coordinate system is chosen to be instantaneously coincident with the "Earth Fixed" coordinate system at time T
STATEV	ENGRAT	Obtains state-vector (satellite or planetary) by evaluating previously developed power series (*)
SVAREQ	SOFORT SOFSEC SOFRT	Evaluates power series to obtain satellite state transition matrix
TDELAY	TDRSS	Computes the transmission time for a radio signal sent from one moving point to another
TDRSS	MEASXX	Computes predicted TDRSS measurements (Measurement types 19-20)
TERRA	PTSTAR	Computes terrain camera orientation angles from stellar camera orientation angles
TERRAS	PTSTAR	Computes terrain camera orientation angles such that the camera axes point due East due North and vertically up
TIMARR	SPOOL1	Rearranges input UT1 and Ephemeris time corrections
UNIFD2	EDIT	Reads data tape in "unified" Format

SUBROUTINE Name	CALLED BY	PURPOSE OR FUNCTION
VISIBI	SIMOUT	Computes and prints time of satellite visibility for each station
VXPROD	SELECT	Used for computing the vector cross product C = A * B.
WRITER	PTCMPA	Utility routine for writing on sequential file (37)
WTSPAX	INP600 INTCOD INPCRD	Utility routine for direct access file 41
XFORM	TDRSS GEOCEG	Uses "Inertial 1950.0" power series for satellite state-vector to compute satellite state-vector power series coefficients in same coordinate system as used by STASER

NOTES

1. The computed matrices are of the form of matrices (3.19.13) through (3.9.16) in "A Photogrammetric and Tracking Network Analysis Program, Old Dominion Systems, Inc., October 1973, Contract DAAK 02-72-C-0434".

2. Routines marked "*" are used in the integrator. They are called once per integration step. Routines marked "**" are used in the integrator. They are called once for each term (beyond the second) in the power series expansion. For a 16 term power series expansion, which is normal, these routines are thus called 14 times per integration step.

APPENDIX G

Photonap Common Blocks

G.1 Listing of common blocks used by each subroutine

Routine	Common Blocks
ATMIN	IATMOS
ATMOUT	ATMOS
CAT000	
DARITE	TSPARM, SOLFIL
DAYHMS	
DBREAD	TSPARM
DBURN	INTCMF, INTCMO, CDEBUG, BURNS
DEFALT	COMSOL, ASPARM, PSPARM, GENCOM, CONMET, ICONST, EXTCM, EARTH, BURNS, DRSGA, SOLDRG
DENS	IATMOS
DGRITE	FOTGNO
DGRITS	FOTGNO
DRAGU	INTCMO, DRSGB
DRAVAR	INTCMG, INTCMO, INTCMI
DREDIT	COMSOL, ACINFO, IONUMB, GENCOM, GPCOM, CDEBUG, ICONST, CWORK, TSPARM
DUMDUM	
EDIT	FOTGND, COMSOL, STINFO, GENCOM, EXTCM, COVAR, CDEBUG, CWORK, IONUMB, ICONST
EDITDT	COMSOL, TSEDIT, ASPARM, ACINFO, PSPARM, GPCOM, ATMOS, DRSGA
EIGN	
ENBVAR	INTCMF, INTCMO, INTCMI, GENCOM
ENGRAT	TIMING, INTCMF, INTCMO, INTCMI, POWER, POTREC, CDEBUG, EXTCM, GENCOM, IONUMB, AJPL, SEROUT

Routine	Common Blocks
ENROOT	
EXPAND	INTCMF, INTCMO, INTCMI, AJPL, POWER
FINALP	GENCOM, TYLE, TSPARM, ASPARF, PSPRMF, CWORK, IONUMB, STINFO, FOTGND, EARTH, COVAR
FINDOG	MISCOM, INTCMF, INTCMO, INTCMI, POTREC, TSPARM, CDEBUG, GENCOM
FIREAD	TSPARM
FLREAD	TSPARM
PHOTO	INROOT
GENFIL	COMSOL, TSPARM, TSEDIT, ASPARM, ACINFO, PSPARM, STINFO, GENCOM, CDEBUG, CWORK, IONUMB, ICONST, GPCOM, TIMING, INROOT, AJPL, EXTCM, POTREC, EARTH, BURNS, COVAR, SOLDRG, DRSGA
GENBUG	COMSOL, ASPARM, PSPARM, CWORK, ICONST, GPCOM, EXTCM, BURNS, SOLDRG
GEOCEG	CMEASR, PARSOM, EXTCM, RSUMR, EARTH, PRTLB, PRTEMP
GEOS	STINFO, IONUMB
GETTAP	CETBL2, INTCMO, CETBL9, REC3
GPRSM2	IONUMB, STINFO
GROUND	FOTO, EARTH
HOPRFT	PRTLB, CMEASR, RSUMR, XPNDR
IEMCOL	COMSOL
IEMSET	COMSOL
IEMVAL	COMSOL
INPCRD	TSPARM, ASPARM, CDEBUG, IONUMB, ICONST, EXTCM, BURNS, INPCMA
INP100	COMSOL, GENCOM, IONUMB, ICONST, GPCOM, EXTCM, EARTH, ATMOS, INPCMA
INP200	COVAR, ACINFO, COMSOL, ICONST, POTREC, EXTCM, BURNS, INPCMA, DRSGA

Routine	Common Blocks
INP300	STINFO, FCONST, INPCMA
INP600	TSPARM, TSEDIT, ASPARM, ACINFO, PSPARM, COMSOL, CWORK, POTREC, GPCOM, INROOT, BURNS, INPCMA, DRSGA
INP700	COMSOL, GENCOM, FCONST, INROOT, INPCMA
INPBUG	TSPARM, TSEDIT, ASPARM, ACINFO, PSPARM, STINFO, TYLE, COMSOL, ICONST
INTCOD	INROOT, TSPARM, TSEDIT, CWORK, GENCOM, GPCOM, POTREC
INTEG	BURNS, DRSGB, CWORK, INTCMF, INTCMO, INTCMI, EXTCM, CINTEG
INTERP	TIMING
INTGA	POTREC, TIMING, TSPARM, EXTCM, INTCMF, INTCMO, INTCMI, POWER
INTP1	TIMING
INTP2	TIMING
INTPRT	
INVERT	SOLCOM, TSSOLV, SOLFIL
INVPRE	SOLFIL, FOTGND, SOLCOM, IONUMB, CWORK, TSSOLV, TSPARM
INVRTB	SOLCOM, TSSOLV, SOLFIL, GENCOM, CWORK, STINFO
INVRTC	SOLCOM, TSSOLV, SOLFIL, GENCOM, CWORK, STINFO
INVSYM	
INVSYS	
JACHIA	TSPARM
JULDAY	
KEPLER	
KICKER	SEROUT, CDEBUG, CINTEG, CWORK, CONMET, GENCOM, EXTCM, IONUMB, PARTY, POTREC, POWER, INTCMF, INTCMO, INTCMI, CETBL1
KMNCON	EARTH, AJPL, CPSYST, KMAN1, KMAN2, KMAN3, KMAN4, KMANI, CINTEG, EXTCM, CWORK

Routine	Common Blocks
KMNEVA	KMAN1, KMAN2, EXTCM
KMNIDE	
KMNINI	GENCOM, CWORK, GPSYST, KMAN1, KMAN2, KMANI, KMAN4, STINFO, IONUMB
KMNINV	KMAN1
KMNMP1	
KMNMP2	
KMNMP3	
KMNOUT	GPSYST, KMAN2, KMAN4, KMANI, GENCOM
KMNRAN	
KMNSIM	GPSYST, KMANI, KMAN4, GENCOM
KMNSM2	KMAN2, KMANI, GENCOM
MASCON	INTCMF, INTCMO, INTCMI, CWORK, POTREC
MATINV	
MATZEV	INTCMO, INTCMI
MATZEX	INTCMF, INTCMO, INTCMI
MAT23	
MAT31	
MAT32	
MAT33	
MAT3T3	
MAT99	
MEASUR	EXTCM, CMEASR, CONMET, EARTH, RSUMR, GENCOM, CINTEG, PRTEMP, PRTLB, AJPL
MEASXX	CMEASR, EARTH, RSUMR, XPNDR, GENCOM, CINTEG, PRTEMP, PRTLB, AJPL
NBDNEX	INTCMF, INTCMO, INTCMI

Routine	Common Blocks
NBDPEX	INTCMF, INTCMO, INTCMI
NEWJPL	INTCMF, INTCMO, INTCMI, CETBL1, CETBL2, CETBL4
NUTION	TIMING
O4DATE	AJPL
OCCULT	INTCMF, INTCMO, INTCMI
OFDATE	AJPL
OLDMAN	GENCOM, IONUMB, OVRLAY
PAGE	GENCOM
PFSOLV	EXTCM, POWER
PFVARY	INTCMF, INTCMO, INTCMI, POWER
PLATEC	FOTO
POTSOL	
POTVAR	INTCMF, INTCMO, INTCMI, AJPL
PREINT	CWORK, CINTEG, TSPARM, EARTH, IONUMB, EXTCM, AJPL, DRSGB
PREPAR	GENCOM, EARTH, TIMING, CDEBUG, CWORK, CINTEG, BURNS, DRSGB, PARTY, POWER, EXTCM, IONUMB, OVRLAY, COVAR
PRTBG1	CWORK
PRTBG2	CWORK, GENCOM, STINFO, IONR
PRTBG3	CINTEG, RSUMR, PRTLB
PRTIAL	CDEBUG, CMEASR, CONMET, CWORK, CINTEG, BURNS, DRSGB, PARTY, SDP, AJPL, POTREC, POWER, GENCOM, EXTCM, IONUMB, STINFO, TSPARM, EARTH, RSUMR, XPNDR, IONR, OVRLAY, TIMING, FOTGND, PRTLB, PRTEMP
PTCMPA	PRTEMP, STINFO, CWORK, GENCOM
PTINIT	STINFO, PRTEMP, CWORK, CINTEG, GENCOM, EXTCM, TSPARM, EARTH, IONR, RSUMR
PTSTAR	PRTEMP, PRTLB, CWORK, CINTEG, AJPL, EXTCM, IONUMB, EARTH, TIMING, FOTO, FOTGND

Routine	Common Blocks
RAN601	
READE	CETBL2, CETBL5, CETBL1, INTCMO, CETBL4, CETBL9
READER	
RECCOF	INTCMF, INTCMO, INTCMI
RECPOT	INTCMF, INTCMO, INTCMI, CWORK, POTREC, GENCOM, AJPL
REDISK	TSPARM
REDSKK	TSPARM, TSSOLV
REDSKM	TSSOLV, SOLCOM, SOLFIL
REFRCT	EARTH, GENCOM, CMEASR, CONMET
REPRT2	
REREAD	TSPARM
RESID	TSPARM, CWORK, STINFO, GENCOM, CDEBUG, IONUMB
ROOTDT	TIMING, STINFO, TYLE, GENCOM, CDEBUG, MISCOM, CWORK, IONUMB, EXTCM, EARTH, ICONST, POTREC, FCONST, CONMET, COVAR, FOTGND, TSPARM, IONR
ROTATE	
ROTINT	OVRLAY, STINFO, EARTH
ROTPAR	EARTH, PRTLB
ROTVFD	EARTH, PRTLB
ROT1	
RSUM	RSUMR, CMEASR, PRTLB
SELECT	GPSYST, KMANI, KMAN2
SIGWT	
SIMOUT	INROOT, CWORK, STINFO, GENCOM, IONUMB
SKRIV	
SOFORT	GENCOM, CDEBUG, PRTLB, CINTEG, EXTCM, SDP, POTREC, BURNS, IONUMB, PARCOM, POWER, AJPL, MISCOM

Routine	Common Blocks
SOFRT	GENCOM, CDEBUG, CINTEG, EXTCM, IONUMB, FINCOM, AJPL, COVAR, MISCOM
SOFSEC	GENCOM, CDEBUG, PRTLB, CINTEG, EXTCM, SDP, POTREC, IONUMB, PARCOM, PARTY, AJPL, MISCOM
SOLAWA	CWORK
SOLREC	
SOLVDT	SOLCOM, TSSOLV, SOLFIL
SOLVER	SOLCOM, FOTGND, TSPARM, TSSOLV, IONUMB, GENCOM, CWORK, CDEBUG, SOLFIL
SOLVFU	SOLCOM, TSSOLV, IONUMB, GENCOM, CWORK
SOLV1	SOLCOM, TSPARM, TSSOLV, GENCOM, SOLFIL
SOLV2	SOLCOM, TSPARM, TSSOLV, GENCOM, CWORK, COVAR, SOLFIL
SOLV3	GENCOM, SOLFIL
SOREAD	CWORK, SOLFIL
SORITE	CWORK, SOLFIL
SPODT	SP1COM, SP2COM, SP3COM
SPOLCD	CONMET, TIMING, INROOT, FOTGND, TSPARM, STINFO, GENCOM, IONUMB, ICONST, CWORK, EARTH, IONR, SP1COM, SP2COM, SP3COM
SPOOL1	TIMING, IONUMB, SP1COM
SPOOL2	FOTGND, GENCOM, IONUMB, SP2COM
SPOVEL	
SPO100	CONMET, POTREC, INROOT, TSPARM, STINFO, TYLE, GENCOM, IONUMB, CDEBUG, CWORK, SP3COM
SP02A3	PRTLB, PARCOM, RSUMR, EXTCM, CMEASR, EARTH, PRTEMP
STARAN	FOTO
STARPA	
STARPI	

Routine	Common Blocks
STASER	EARTH
STATEV	INTCMO, INTCMI
SVAREQ	
TDELAY	RSUMR
TDRSS	CMEASR, PRTLB, PARCOM, EXTCM, RSUMR, XPNDR, EARTH, GENCOM
TERRA	FOTO
TERRAS	FOTO, PRTLB
TIMARR	TIMING
UNIFD2	STINFO, IONUMB
VISIBI	GENCOM, IONUMB, STINFO
VXPROD	
WRITER	
WTSPAX	TSPARM, TSEDIT, IONUMB
XFORM	EXTCM
IONOSF	CMEASR, PRTLB, STINFO, IONR, IONTM
ION	CMEASR, IONR
BETA	
READU	IONR
DKSICO	
DKGK	
GK	
MAGFIN	
SICOJT	

G.2 Listing of subroutines utilizing each common block

Common Block	Routines Used in
ACINFO	DREDIT, EDITDT, GENFIL, INP200, INP600, INPBUG
AJPL	ENGRAT, EXPAND, GENFIL, MEASUR, MEASXX, O4DATE OFDATE, POTVAR, PREINT, PRTIAL, PTSTAR, RECPOT, SOFORT, SOFRT, SOFSEC, KMNCON
ASPARF	FINALP
ASPARM	DEFALT, EDITDT, GENFIL, GENBUG, INPCRD, INP600, INPBUG
ATMOS	ATMOUT, EDITDT, INP100
BURNS	DBURN, DEFALT, GENFIL, GENBUG, INPCRD, INP200, INP600, INTEG, PREPAR, PRTIAL, SOFORT
CDEBUG	DBURN, DREDIT, EDIT, ENGRAT, FINDOG, GENFIL, INPCRD, KICKER, PREPAR, PRTIAL, RESID, ROOTDT, SOFORT, SOFRT, SOFSEC, SOLVER, SPO100
CETBL1	KICKER, NEWJPL, READE
CETBL2	GETTAP, NEWJPL, READE
CETBL4	NEWJPL, READE
CETBL5	READE
CETBL9	GETTAP, READE
CINTEG	INTEG, KICKER, MEASUR, MEASXX, PREINT, PREPAR, PRTBG3, PRTIAL, PTINIT, PTSTAR, SOFORT, SOFRT, SOFSEC, KMNCON
CMEASR	GEOCEG, HOPRFT, MEASUR, MEASXX, PRTIAL, REFRCT, RSUM, SPO2A3, TDRSS, IONOSF, ION
COMSOL	DEFALT, DREDIT, EDIT, EDITDT, GENFIL, GENBUG, IEMCOL, IEMSET, IEMVAL, INP100, INP200, INP600, INP700, INPBUG
CONMET	DEFALT, KICKER, MEASUR, PARTIAL, REFRCT, ROOTDT, SPOLCD, SPO100
COVAR	EDIT, FINALP, GENFIL, INP200, PREPAR, ROOTDT, SOFRT, SOLV2

Common Block	Routines Used in
CWORK	SOLVFU, SP0100, DREDIT, EDIT, FINALP, GENFIL, GENBUG, INP600, INTCOD, INTEG, INVPRE, INVRTB, INVRTC, KICKER, MASCON, PREINT, PREPAR, PRTBG1, PRTBG2, PRTIAL, PTCMPA, PTINIT, PTSTAR, RECPOT, RESID, ROOTDT, SIMOUT, SOLVER, SOREAD, SOLFIL, SPOLCD, KMNCON, KMNINI, SOLAWA, SOLV2
DRSGA	DEFALT, EDITDT, INP200, INP600, GENFIL
DRSGB	INTEG, PREINT, PREPAR, PRTIAL, DRAGU
EARTH	DEFALT, FINALP, GENFIL, GEOCEG, GROUND, INP100, MEASUR, MEASXX, PREINT, PREPAR, PRTIAL, PTINIT, PTSTAR, REFRCT, ROOTDT, ROTINT, ROTPAR, ROTVFD, SPOLCD, SPO2A3, STASER, TDRSS, KMNCON
EXTCM	DEFALT, EDIT, ENGRAT, GENFIL, GENBUG, GEOCEG, INPCRD, INP100, INP200, INTEG, INTGA, KICKER, MEASUR, PFSOLV, PREINT, PREPAR, PRTIAL, PTINIT, PTSTAR, ROOTDT, SOFORT, SOFRT, SOFSEC, SPO2A3, TDRSS, XFORM, KMNCON, KMNEVA
FCONST	INP300, INP700, ROOTDT
FINCOM	SOFRT
FOTGND	DGRITE, DGRITS, EDIT, FINALP, INVPRE, PRTIAL, PTSTAR, ROOTDT, SOLVER, SPOLCD, SPOOL2
FOTO	GROUND, PLATEC, PTSTAR, STARAN, TERRA, TERRAS
GENCOM	SOLV3, SOLVFU, SPOOL2, SP0100, VISIBI, KMNINI, KMNOUT, KMNSIM, KMNSM2, SOLV1, SOLV2, DEFALT, DREDIT, EDIT, ENBVAR, ENGRAT, FINALP, FINDOG, GENFIL, INP100, INP700, INTCOD, INVRTB, INVRTC, KICKER, MEASUR, MEASXX, OLDMAN, PAGE, PREPAR, PRTBG2, PRTIAL, PTCMPA, PTINIT, RECPOT, REFRCT, RESID, ROOTDT, SIMOUT, SOFORT, SOFRT, SOFSEC, SOLVER, SPOLCD, TDRSS
GPCOM	DREDIT, EDITDT, GENFIL, GENBUG, INP100, INP600, INTCOD
GPSYST	KMNCON, KMNINI, KMNOUT, KMNSIM, SELECT
IATMOS	ATMIN, DENS
ICONST	DEFALT, DREDIT, EDIT, GENFIL, GENBUG, INPCRD, INP100, INP200, INPBUG, ROOTDT, SPOLCD

Common Block	Routines Used in
INPCMA	INPCRD, INP100, INP200, INP300, INP600, INP700
INROOT	PHOTO, GENFIL, INP600, INP700, INTCOD, SIMOUT, SPOLCD, SPO100
INTCMF	DBURN, DRAVAR, ENBVAR, ENGRAT, EXPAND, FINDOG, INTEG, INTGA, KICKER, MASCON, MATZEX, NBDNEX, NBDPEX, NEWJPL, OCCULT, PFVARY, POTVAR, RECCOF, RECPOT
INTCMI	DRAVAR, ENBVAR, ENGRAT, EXPAND, FINDOG, INTEG, INTGA, KICKER, MASCON, MATZEV, MATZEX, NBDNEX, NBDPEX, NEWJPL, OCCULT, PFVARY, POTVAR, RECCOF, RECPOT, STATEV
INTCMO	DBURN, DRAVAR, ENBVAR, ENGRAT, EXPAND, FINDOG, GETTAP, INTEG, INTGA, KICKER, MASCON, MATZEV, MATZEX, NBDNEX, NBDPEX, NEWJPL, OCCULT, PFVARY, POTVAR, READE, RECCOF, RECPOT, STATEV, DRAGU
IONR	PRTBG2, PRTIAL, PTINIT, ROOTDT, SPOLCD, IONOSF, ION, READU
IONTM	IONOSF
IONUMB	KMNINI, SOLVFU, SPOOL1, SPOOL2, SPO100, DREDIT, EDIT, ENGRAT, FINALP, GENFIL, GEOS, GPRSM2, INPCRD, INP100, INVPRE, KICKER, OLDMAN, PREINT, PREPAR, PRTIAL, PTSTAR, RESID, ROOTDT, SIMOUT, SOFORT, SOFRT, SOFSEC, SOLVER, SPOLCD, UNIFD2, VISIBI, WTSPAX
KMANI	KMNCON, KMNINI, KMNOUT, KMNSIM, KMNSM2, SELECT
KMAN1	KMNCON, KMNEVA, KMNINI, KMNINV
KMAN2	KMNCON, KMNEVA, KMNINI, KMNOUT, KMNSM2, SELECT
KMAN3	KMNCON
KMAN4	KMNCON, KMNINI, KMNOUT, KMNSIM
MISCOM	FINDOG, ROOTDT, SOFORT, SOFRT, SOFSEC
OVRLAY	OLDMAN, PREPAR, PRTIAL, ROTINT
PARCOM	GEOCEG, SOFORT, SOFSEC, SPO2A3, TDRSS
PARTY	KICKER, PREPAR, PRTIAL, SOFSEC

Common Block	Routines Used in
POTREC	ENGRAT, FINDOG, GENFIL, INP200, INP600, INTCOD, INTGA, KICKER, MASCON, PRTIAL, RECPOT, ROOTDT, SOFORT, SOFSEC, SPO100
POWER	ENGRAT, EXPAND, INTGA, KICKER, PFSOLV, PFVARY, PREPAR, PRTIAL, SOFORT
PRTEMP	GEOCEG, MEASUR, MEASXX, PRTIAL, PTCMPA, PTINIT, PTSTAR, SPO2A3
PRTLB	GEOCEG, HOPRFT, MEASUR, MEASXX, PRTBG3, PRTIAL, PTSTAR, ROTPAR, ROTVFD, RSUM, SOFORT, SOFSEC, SPO2A3, TDRSS, TERRAS, IONOSF
PSPARM	DEFALT, EDITDT, GENFIL, GENBUG, INP600, INPBUG
PSPRMF	FINALP
REC3	GETTAP
RSUMR	GEOCEG, HOPRFT, MEASUR, MEASXX, PRTBG3, PRTIAL, RSUM, SPO2A3, TDELAY, TDRSS, PTINIT
SDP	PRTIAL, SOFORT, SOFSEC
SEROUT	ENGRAT, KICKER
SOLCOM	INVERT, INVPRE, INVRTB, INVRTC, REDSKM, SOLVDT, SOLVER, SOLV1, SOLV2, SOLVFU
SOLDRG	DEFALT, GENFIL, GENBUG
SOLFIL	DARITE, INVERT, INVPRE, INVRTB, INVRTC, REDSKM, SOLVDT, SOLVER, SOREAD, SORITE, SOLV1, SOLV2, SOLV3
SP1COM	SPOLCD, SPODT, SPOOL1
SP2COM	SPOLCD, SPODT, SPOOL2
SP3COM	SPOLCD, SPODT, SPO100
STINFO	EDIT, FINALP, GENFIL, GEOS, GPRSM2, INP300, INPBUG, INVRTB, INVRTC, PRTBG2, PRTIAL, PTCMPA, PTINIT, RESID, ROOTDT, ROTINT, SIMOUT, SPOLCD, UNIFD2, VISIBI, IONOSF, KMNINI, SPO100
TIMING	ENGRAT, GENFIL, INTERP, INTGA, INTP1, INTP2, NUTION, PREPAR, PRTIAL, PTSTAR, ROOTDT, SPOLCD, TIMARR, SPOOL1

Common Block	Routines Used in
TSEDIT	EDITDT, GENFIL, INP600, INPBUG, INTCOD, WTSPAX
TSPARM	DARITE, DBREAD, DREDIT, FINALP, FINDOG, FIREAD, FLREAD, GENFIL, INPCRD, INP600, INPBUG, INTCOD, INTGA, INVPRE, PREINT, PRTIAL, PTINIT, REDISK, REDSKK, REREAD, RESID, ROOTDT, SOLVER, SPOLCD, WTSPAX, JACHIA, SOLV1, SOLV2, SP0100
TSSOLV	INVERT, INVPRE, INVRTB, INVRTC, REDSKK, REDSKM, SOLVDT, SOLVER, SOLV1, SOLV2, SOLVFU
TYLE	FINALP, INPBUG, ROOTDT, SP0100
XPNDR	HOPRFT, MEASXX, PRTIAL, TDRSS